DESIGNING FRACTIONAL FACTORIAL SPLIT-PLOT EXPERIMENTS
USING INTEGER PROGRAMMING

by

Shay Randall Capehart

A Dissertation Presented in Partial Fulfillment
of the Requirements for the Degree
Doctor of Philosophy

ARIZONA STATE UNIVERSITY

August 2008

DEFENSE TECHNICAL INFORMATION CENTER

Information for the Defense Community

Month Day Year

DTIC® has determined on 09 09 2008 that this Technical Document has the Distribution Statement checked below. The current distribution for this document can be found in the DTIC® Technical Report Database.

[X] **DISTRIBUTION STATEMENT A.** Approved for public release; distribution is unlimited.

[ ] © **COPYRIGHTED.** U.S. Government or Federal Rights License. All other rights and uses except those permitted by copyright law are reserved by the copyright owner.

[ ] **DISTRIBUTION STATEMENT B.** Distribution authorized to U.S. Government agencies only. Other requests for this document shall be referred to controlling office.

[ ] **DISTRIBUTION STATEMENT C.** Distribution authorized to U.S. Government Agencies and their contractors. Other requests for this document shall be referred to controlling office.

[ ] **DISTRIBUTION STATEMENT D.** Distribution authorized to the Department of Defense and U.S. DoD contractors only. Other requests shall be referred to controlling office.

[ ] **DISTRIBUTION STATEMENT E.** Distribution authorized to DoD Components only. Other requests shall be referred to controlling office.

[ ] **DISTRIBUTION STATEMENT F.** Further dissemination only as directed by controlling office or higher DoD authority.

Distribution Statement F is also used when a document does not contain a distribution statement and no distribution statement can be determined.

[ ] **DISTRIBUTION STATEMENT X.** Distribution authorized to U.S. Government Agencies and private individuals or enterprises eligible to obtain export-controlled technical data in accordance with DoDD 5230.25.

DESIGNING FRACTIONAL FACTORIAL SPLIT-PLOT EXPERIMENTS
USING INTEGER PROGRAMMING

by

Shay Randall Capehart

has been approved

July 2008

Graduate Supervisory Committee:

Douglass Montgomery, Chair
Murat Kulahci
Ahmet Keha
Connie Borror

ACCEPTED BY THE GRADUATE COLLEGE

ABSTRACT

Split-plot designs are commonly used in industrial experiments when there are hard-to-change and easy-to-change factors. Due to the number of factors and resource limitations, it is more practical to run a fractional factorial split-plot (FFSP) design. These designs are variations of the fractional factorial (FF) design, with the restricted randomization structure to account for the whole plots and subplots. We begin by discussing the formulation of FFSP designs using integer programming (IP) to achieve various design criteria. We specifically look at the maximum number of clear two-factor interactions and variations on this criterion. By making restrictions on some of the general linear constraints, we are able to customize the alias structure of these FFSP designs. Additional constraints allow for the generation of blocked FFSP designs that are shown to meet performance standards shown in today's literature. By generalizing the model formulation, we show how designs for numerous stages can be generated. In addition, we explore using a genetic algorithm heuristic to search for split-plot designs from a candidate matrix of factor effects generated using the Kronecker product.

This dissertation is dedicated to Karl Frank.

ACKNOWLEDGEMENTS

I wish to acknowledge the many contributions to this research by my committee members. In particular, I would like to acknowledge my chair Dr. Douglas C. Montgomery and Dr. Murat Kulahci for his continued guidance and support throughout my three-year stay at ASU, Dr. Murat Kulahci for his insight in exploring split-plot designs to kick-start my research, and Dr. Ahmet Keha for his Operational Research expertise. Finally, I would like to acknowledge the support of the United States Air Force, in particular the ROTC Det 25 staff for their support and assistance.

TABLE OF CONTENTS

LIST OF TABLES

Table Page

CHAPTER 1: INTRODUCTION

Design of Experiments

Designed experiments offer a systematic approach to study the effects of several factors on process performance. By analyzing the observations from the DOE, one can efficiently determine the factors and interactions that influence one or more response variables. This structured set of analysis often requires three specific statistical assumptions about the observations errors: independence, normality, and constant variance (Montgomery, 2001). The most important of these three is the need for the observations to be independent. Completely randomizing the order of the factor levels satisfies this independence assumption for designed experiments. Often, it can be difficult to change one of the factor levels due to physical or economic restrictions, which limits the ability to run these factor levels in a random order. When this happens, restrictions are placed on the randomization of experimental runs, which results in a split-plot design (SPD). (Box & Jones, 1992)

Split-Plot Designs

In a split-plot design, Box and Jones (1992) call the factors that are restricted hard-to-change factors, and places them within a whole plot. While the other easy-to-change factors are placed within subplots. Each time the factor levels for the whole plot are set, we run all or some of the combinations for the easy-to-change factors within the subplot.

Due to the restrictions on randomization for the whole plot factors, split-plot designs cannot be analyzed using the traditional DOE method. Simple experiments have one

randomization, since all the factors are randomized. For a simple two-factor model, complete randomization of k replications results in the following model structure:

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + \varepsilon_{ijk}$$

$$i = 1,2, \quad j = 1,2,3, \quad k = 1,2, \quad \varepsilon_{ijk} \sim \text{i.i.d. } N\left(0, \sigma_\varepsilon^2\right)$$

Where μ is the overall mean, α_i denotes the effect of factor A, β_j denotes the effect of factor B, $(\alpha\beta)_{ij}$ denotes their interaction, and ε_{ijk} represents the random error associated with the experimental units. However, for a split-plot design, there are two randomizations. The whole-plot factors are randomized according to the whole plot design. Within each whole plot, the subplot factors are randomized independently of the whole plot configuration. For replicated designs, these two randomizations lead to two independent error components, one for the whole plot treatments and another for the subplot treatments. The model structure for a split-plot design, including both error terms, is

$$y_{ijk} = \mu + \alpha_i + \beta_j + (\alpha\beta)_{ij} + b_k + w_{ik} + \varepsilon_{ijk}$$

$$i = 1,2, \quad j = 1,2,3, \quad k = 1,2,$$

$$b_k \sim \text{i.i.d. } N\left(0, \sigma_{\text{block}}^2\right) \quad w_{ik} \sim \text{i.i.d. } N\left(0, \sigma_{\text{WP}}^2\right) \quad \varepsilon_{ijk} \sim \text{i.i.d. } N\left(0, \sigma_\varepsilon^2\right)$$

Where in addition to the previous notation, b_k denotes the random effect associated with the kth block or replication of the whole-plots, w_{ik} denotes the random effect associated with the ith whole-plot within the kth block, and ε_{ijk} denotes the random errors associated with the sub-plot experimental units within the ikth whole-plot.

Goos and Vandebroek (2001) point out that split-plot designs cause a loss in precision in estimating the whole plot coefficients, while increasing the precision in estimation of the sub-plot coefficients and the whole plot by sub-plot interactions. Box and Jones (1992) show that the error variance for the whole-plot and subplot surround that of the error variance for a completely randomized design (CRD), where $\sigma^2_{sub} < \sigma^2_{CRD} < \sigma^2_{whole}$. Similar research showing the increased precision for the subplot treatments is found in Kulahci et al. (2006).

Parker et al. (2006) provide conditions on the structure of split-plot designs where the ordinary least squares (OLS) estimates and generalized least squares (GLS) estimates are equivalent. When dealing with saturated designs or fractional factorials, it is not always possible to maintain orthogonality in the design structure. Loss of orthogonality affects the consistency of our coefficient estimates across the design space (Lin, 1993).

Goos and Vandebroek (2001) describe a method to design SPDs using the *D*-optimality criterion. Due to the split-plot error structure, they use prior information to estimate the ratio between the whole plot error variance to the sub-plot error variance. Using simulations, they show that the *D*-criterion value is a reliable measure for comparing split-plot designs with finite sample size given a fairly good estimate for the variance ratio, such as using the REML estimation method. Goos and Vandebroek point out that one only needs to determine the correct design points to design a good completely randomized design, while designing a good split plot experiment consists of simultaneously choosing the number of whole plots and number of sub-plots within each whole plot. This decision affects the structure of the variance-covariance matrix. As

pointed out by Goos and Vandebroek, SPDs cause a loss in precision in estimating the whole plot coefficients, while increasing the precision in estimation of the sub-plot coefficients and the whole plot by sub-plot interactions.

Fires and Hunter (Fries & Hunter, 1980) first referred to the minimum aberration as criteria for comparing fractional 2^{k-p} designs instead of resolution. Let there exist two 2^{7-2} designs with defining relations **I** = ABCF = BDEG = ACDEFG and **I** = ABCDF = ABCEG = DEFG, both Resolution IV designs. The second design is said to have smaller aberration because it has fewer words of length 4 (Huang, Chen, & Voelkel, 1998). Refer to Chen (1992) for the formal definition of minimum aberration fractional factorial designs.

Fractional factorial designs, or 2^{k-p} factorials, are primarily used in experiments where there are a large number of factors to consider. Fractional factorials of Resolution III are also called saturated designs and only require $k + 1$ runs to investigate the main k factors (Box & Hunter, 1961). Fractional Factorial Split-Plot (FFSP) designs use fractional factorial designs for the subplot structures (Kempthorne, 1998). Bingham and Sitter (1999) and Huang et al. (1998) provide methods for generating minimum aberration fractional factorial split-plot designs and give tables for various combinations of whole plot and subplot factors. Bingham and Sitter (2001) apply the FFSP method to a real industrial experiment to demonstrate how the split-plot design affects estimates, precision, and resource allocation.

Saturated designs are often used in screening experiments in order to reduce a long list of candidate variables to a relatively small number, allowing additional experiments

to be more efficient and require fewer runs. Screening experiment is Phase 0 of the sequential experimental process described by Myers and Montgomery (2002). Once the important variables are identified, Phase 1 consists of developing a first-order model to move the process toward the optimum response value. Phase 2 occurs when the process nears the optimum. Usually, a second order model is constructed during this phase.

Another type of saturated design often used in screening experiments is the Plackett-Burman (PB) design (Plackett & Burman, 1946). PB designs have high D-efficiency, but also complex alias structures (Hamada & Wu, 1992). This complex alias structure makes it extremely difficult to interpret the results (Montgomery, Borror, & Stanley, 1997), because often main effects are either aliased with each other or two-factor interactions.

In blocked designs, the selection of the defining contrasts and effects to be confounded with blocks is very complicated. This same difficulty occurs in fractional designs, including split-plot designs. Franklin and Bailey (1977) provide a procedure to produce fractional factorial designs where certain main effects and interactions are to be estimated, all other effects being negligible. This approach requires a lot of computation due to the exhaustive search; therefore, the approach becomes impractical with large numbers of factors. Liao and Iyer (1999) use a stochastic search method, SEF (sequential elimination of factors), as a modification to the exhaustive approach of Franklin and Bailey. While the SEF algorithm is less computational intensive as the Franklin-Bailey procedure, it sometimes fails to find a solution when the required design is close to a saturated design.

Custom Criteria

Kulahci et al. (2006) look at several design criterions for SPDs, paying close attention to minimum aberration. They consider a sequential process example, where split-plot designs offer great opportunities in increasing the precision in estimating the interactions between the factors in the first step and the factors from within the second step. While minimum aberration focuses on finding the design with the least amount of overall confounding (number of four-factor interactions with the mean), it does not try to maximize the number of clear two-factor interactions. Even this strategy of maximizing the number of clear two-factor interactions, which Wu and Wu (2002) call MaxC2 designs, is not desirable in distinguishing among the different types of two-factor interactions that occur in split-plot designs. Namely, in SPDs, there are two-factor interactions within the whole plot and sub-plot design structures, plus there are two-factor interactions between whole plot factors and sub-plot factors. The later set of interactions is normally the main reason for analysis of sequential processes, where the goal is to study the interactions between the first step (whole plot factors) and second step (sub-plot factors). Kulahci et al. look at three design examples for a two-step process with 7 two-level factors in the first step and 8 two-level factors in the second step. Each design has 64 runs. The first design is the minimum aberration design, while the second and third designs are two 64-run Fractional Factorial Split Plot designs using two different sets of generators. Both of the FFSPs allow the authors to estimate a larger number of clear 2FI. Their examples show that minimum aberration designs do not always provide the maximum amount of information when looking at a 2-step process with a fractional

factorial experiment. They further break down the two FFSPs to point out the greater number of clear two-factor interactions between whole plot and sub-plot factors. Kulahci et al. feel instead of relying on an optimization method to create a design, such as minimum aberration or MaxC2, on should decide the best design based on exactly what information you are trying to gather from the design. While the authors do not provide an algorithm for generating these designs, they do provide two examples that perform better at estimating clear cross-plot interactions.

Linear Programming

In order to create our desired experimental design, we first construct a mathematical model that represents the essence of the problem. Although there are various types of mathematical models, one that is commonly used in optimization is the **linear programming** (LP) model, where the goal is the optimization (minimization or maximization) of a linear **objective function** while meeting a set of linear equality and/or inequality **constraints** or restrictions. George Dantzig is credited for conceiving the LP model during WWII while advising the United States Air Force on developing a planning tool for a deployment, training, and logistics supply program. The model consists of a number of certain quantifiable values to be made called **decision variables** (e.g., x_1, x_2, …, x_n). The objective function provides a quantitative measure of performance to represent a unique set of decision variables. LP models are constructed to fit various types of problems in industry including routing, designs, planning, scheduling, and allocation. One of the assumptions of LP models is *divisibility*, which allows noninteger values for the decision variables. There are many applications where this assumption

presents a limitation. For example, it is often necessary to assign people, machines, or supplies to activities in integer quantities. In these situations, we formulate an Integer Programming (IP) model where all unknown variables are required to be integers, which is simply the IP model with the additional restriction that the decision variables must have integer values. Furthermore, some of these variables may be required to be 0 or 1 (rather than arbitrary integers) and considered binary variables. These type of IP problems are known to be NP-hard (non-deterministic polynomial-time hard) and can become computationally infeasible to solve if the problem becomes too large.

AMPL is a modeling language system that uses various optimization software packages to solve LP models. One such package is CPLEX. CPLEX used advanced optimization algorithms to find solutions to various LP problems, including integer LP problems. The latest version is CPLEX 11.1 available from ILOG.

Summary and Outline of Research

Chapter 2 contains the journal article submitted to *Technometrics* in 2008. During this chapter, we look at creating a fractional factorial split-plot design using integer programming techniques to determine the design generators in order to meet specified goals. We also show how to add constraints to customize the design to meet various aliasing schemes. Instead of working with a traditional design matrix when calculating the alias structure, we demonstrate how to represent the decision variables as binary variables, corresponding to the standard letter-notation commonly used in design of experiments. A collection of examples are provided for 8-, 16-, and 32- run designs. We look at 64-run example from literature to see how our approach allows the user to further

customize which effects and interactions are clear during the analysis. In conclusion, we compare and contrast our split-plot design to those that can be created using a popular statistical software package.

Chapter 3 extends the previous area into blocking FFSP designs. By using IP to formulate the problem, we allow the user to find *D*-optimal designs that also optimize the specified user criteria for the objective function. We use goal programming to look at a series of criterion in the objective function, including number of clear main effects, control/noise effects, two-factor interactions, and even consider interactions that are tested against the whole-plot error component. Our results are compared to those from literature for screening experiments and robust parameter designs. Chapter 3 is also in journal format, and will be submitted in late 2008.

Chapter 4 focuses on manufacturing settings that involve multiple stages in the production line. By considering each subsequent stage in the production line as a sub-design, we can look at the overall design as multiple split-plot design. That is, a design for a three-staged assembly line would be considered a split-split-plot design, with the first stage as the whole plot design, the second stage as the split-plot sub-design, and the final stage as the split-split-plot sub-design. We modify the IP model from our previous research to create a FFSP design for a three-staged manufacturing problem and demonstrate how this approach can be applied systems with more than three stages.

Before focusing on generating split-plot designs using IP models, we proposed using genetic algorithms to search through a candidate set of matrix columns to generate various split-plot designs. Although we changed direction for our final research,

Chapter 5 provides some promising initiatives for future research. Using the Kronecker products operator, we are able to generate an $n \times n$ matrix that represents all possible orthogonal experimental runs for a *n*-run design. We explain how to partition this matrix into two sets: the candidate set for whole-plot factors, and the candidate set for subplot factors. Creating a split-plot design from this matrix becomes a combinatorial problem of selecting the proper WP and SP factors from their respective candidate set to meet the desired performance goal. A simple genetic algorithm (GA) is proposed to search through the set of candidate columns for the split-plot design factors; however, without preprocessing on the candidate sets, the GA takes considerable computational time, especially since we are calculating the alias matrix for each proposed design. In order to reduce the size of the problem, we first the idea of eliminating the *basic factors* columns from consideration in the selection process. For Resolution IV designs, we can also eliminate several other columns from the design, bringing down the computation time considerably. We present results for designs of various factor and run sizes, including a the 64-run split-plot design explored in Kulahci et al. (2006).

CHAPTER 2: DESIGNING FRACTIONAL FACTORIAL SPLIT-PLOT EXPERIMENTS USING INTEGER PROGRAMMING

Introduction

Manufacturing often involves a product going through a production process where the output characteristics of this product reflect the effect of factors that are hard-to-change and factors that are easy-to-change. Determining these effects involve a systematic approach to a designed experiment. Many complex designed experiments are used in real-world systems due to the nature of the system or to reduce the cost to determine process effects. The runs in most designed experiments are completely random; however, this can be very expensive or even impractical when the process involves hard-to-change factors. For that, a split plot design (SPD) where several experiments involving easy-to-change factors are run at fixed levels of hard-to-change factors can be employed. The SPD structure usually requires significantly less resources than other designs used for examining systems involving factors that are hard-to-change. The goal is often to quickly and efficiently build a SPD based on the experimental goals and requirements, e.g. to be able to clearly estimate certain effects.

Fractional factorial designs, or 2^{n-k} factorials, are primarily used in experiments where there are a large number of factors to consider. Fractional factorials of Resolution III only require $k + 1$ runs to investigate the k factors (Box & Hunter, 1961). Fractional Factorial Split-Plot (FFSP) designs use fractional factorial designs for the whole plot/subplot structures and are orthogonal (Kempthorne, 1998). Bingham and Sitter (1999) and Huang et al. (1998) provide methods for generating minimum aberration

(MA) FFSP designs and give tables for various combinations of whole plot and subplot factors. The MA design criterion (Fries & Hunter, 1980) provides a way to distinguish between designs of maximum resolution. Bingham and Sitter (2001) apply the FFSP method to a real industrial experiment to demonstrate how the restriction on randomization in SPD affects estimates, precision, and resource allocation. Kulahci (2007) uses the Kronecker product operation to create a flexible matrix of blocked fractional factorial designs, which could be used as FFSP designs, for various design criteria.

While the analysis of split-plot models often involve generalized least squares (GLS) and restricted maximum likelihood (REML) to estimate the effects, we can use the equivalence between ordinary least squares (OLS) and GLS for our first-order models (Bisgaard, 2000). Vining, Kowalski, and Montgomery (2005) provide conditions which make this equivalence hold for any order model. In addition to this equivalence property, Goos and Vanderbroek (2001) prove that two-level fractional factorial designs arranged in orthogonal blocks, such as FFSP designs, are *D*-optimal when estimating a model consisting of main effects and unconfounded interaction effects. This *D*-optimality refers to minimizing the determinant of the covariance matrix (Goos, 2006).

Several authors have already made tables of FFSP designs when using common design criteria. Bingham and Sitter (2001) presented tables for 8-, 16-, and 32-run FFSP designs using the MA design criterion. Another criterion is the maximum number of clear two-factor interactions (2FIs), which can be shown to have advantages over MA designs in some situations (Chen, Sun, & Wu, 1993). Wu and Wu (2002) refer to these

designs as MaxC2 designs and discuss the rules regarding this criterion for fractional factorials. Yang et. al. (2006) classify the 2FIs in a FFSP design into three categories: WP2FI, SP2FI and WS2FI, where WP2FI and SP2FI refer to 2FIs involving two WP or SP factors respectively, and WS2FI means a 2FI between one WP factor and one SP factor. Although not limited to balanced FFSP designs, Goos and Vandebroek (2001; 2004) have studied optimal split-plot designs using the *D*-optimality criterion. Kulahci et al. (2006) present a compelling argument for custom FFSP designs not simply based on a single criterion, but based on the alias structure, including estimating certain types of clear two-factor interactions. Cheng et al. (1999) consider the model robustness by looking at the *estimation capacity* and the expected number of *suspect* 2FIs. While they conclude that MA designs are highly efficient with regards to these two criteria, if the number of active 2FIs is large, other designs could prove more useful. Jones and Goos (2007) describe an algorithm for finding tailor-made *D*-optimal FFSP designs that can handle flexible choices of sample size, both continuous and categorical factors, and may include interaction terms of any order.

By modeling the system as an IP, we will show how the user can create customized FFSP designs to isolate main effects and 2FIs. The user can add specific constraints determine the Resolution of the design, number of clear WP and SP effects, and even isolate certain 2FIs associated with the individual main effects.

Model Representation

Let's consider a FFSP design with $2^{(n_1+n_2)-(k_1+k_2)}$ runs, $2^{(n_1-k_1)}$ *whole plots* (WP) and $2^{(n_2-k_2)}$ *subplot* (SP) runs within each whole plot. There are n_1 WP and n_2 SP factors,

along with k_1 WP and k_2 SP fractional generators. In such designs, the first $n_1 - k_1$ and $n_2 - k_2$ factors for the WP and SP can be considered *basic factors* and represented by single letters (Franklin & Bailey, 1977). The rest of the factors are represented by the interaction of these basic factors. There exists a set of $2^{(n_1+n_2)-(k_1+k_2)} - 1$ letter groups formed using the letter group notation, which can be arranged using Yates order as follows:

[**A B AB C AC BC ABC** ...]

As indicated above, in this notation there will be $(n_1 + n_2) - (k_1 + k_2)$ columns represented with a single letter, whereas the rest of the columns in which the remaining factors can be allocated are represented as the combinations of the $(n_1 + n_2) - (k_1 + k_2)$ single letters.

Using multiples of 2, the first $p_1 = 2^{(n_1-k_1)} - 1$ letter groups correspond to the possible choices for WP factors. For example, if $n_1 = 6$ and $k_1 = 2$, then the first four WP main effects are [**A B C D**] and the 2 WP fractional generators are chosen from the remaining 11 combination letter groups [**AB AC BC ABC AD** ...]. The other $p_2 = 2^{(n_1+n_2)-(k_1+k_2)} - 2^{(n_1-k_1)}$ letter groups correspond to the choices for SP effects, with $n_2 - k_2$ single letter factors and k_2 SP level design generators.

Although using letter notation to represent factors of a FFSP design is useful in a classroom setting, for computer modeling purposes, we'd like to represent these letter groups using numeric values. We can accomplish this by using a reverse form of binary conversion based upon each single letter, i.e. reading the binary number from left to right.

For example, a 32-run FFSP design will have 5 single letters, **ABCDE**, forming a 5-digit binary number. Thus, looking at letter group **ABD**, we have **AB_D_** = 11010 in binary notation, which can be converted to a numeric value of 11 using reverse binary conversion. Table 1 displays the 31 letter groups and their corresponding numeric value for a 32-run FFSP design using this form of reverse binary conversion.

Table 1: Letter notations for 32-run FFSP design

Numeric Value	Letter-group	**A**	**B**	**C**	**D**	**E**
1	**A**	1	0	0	0	0
2	**B**	0	1	0	0	0
3	**AB**	1	1	0	0	0
4	**C**	0	0	1	0	0
5	**AC**	1	0	1	0	0
6	**BC**	0	1	1	0	0
7	**ABC**	1	1	1	0	0
8	**D**	0	0	0	1	0
9	**AD**	1	0	0	1	0
10	**BD**	0	1	0	1	0
11	**ABD**	1	1	0	1	0
12	**CD**	0	0	1	1	0
13	**ACD**	1	0	1	1	0
14	**BCD**	0	1	1	1	0
15	**ABCD**	1	1	1	1	0
16	**E**	0	0	0	0	1
17	**AE**	1	0	0	0	1
18	**BE**	0	1	0	0	1
19	**ABE**	1	1	0	0	1
20	**CE**	0	0	1	0	1
21	**ACE**	1	0	1	0	1
22	**BCE**	0	1	1	0	1
23	**ABCE**	1	1	1	0	1
24	**DE**	0	0	0	1	1
25	**ADE**	1	0	0	1	1
26	**BDE**	0	1	0	1	1
27	**ABDE**	1	1	0	1	1
28	**CDE**	0	0	1	1	1
29	**ACDE**	1	0	1	1	1
30	**BCDE**	0	1	1	1	1
31	**ABCDE**	1	1	1	1	1

For FFSP designs, we are often interested in estimating main effects and two-factor interactions (2FIs), so it is interesting to note that this set is closed under multiplication. Thus, all 2FIs of the k letter groups can be simplified back into the same set. For example, if factor **D** is selected and one of the fractional generators is represented by the letter group **BCD**, their 2FI corresponds to the (**D**)(**BCD**)=**BC** letter group. To calculate this, we add each digit of the letter groups, modulus 2.

$$\begin{array}{rl} 0\ 0\ 0\ 1\ 0 & = \mathbf{D\ (8)} \\ +\underline{0\ 1\ 1\ 1\ 0} & \underline{= \mathbf{BCD\ (14)}} \\ 0\ 1\ 1\ 0\ 0 & = \mathbf{BC\ (6)} \end{array}$$

Table 2 represents all 465 2FIs and the letter group they correspond to using this numbering scheme for a 32-run FFSP design.

Table 2: Numeric values for two-factor interactions (Row x Column) of letter groups from a 32-run FFSP design

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	3	2	5	4	7	6	9	8	11	10	13	12	15	14	17	16	19	18	21	20	23	22	25	24	27	26	29	28	31	30
2	-	1	6	7	4	5	10	11	8	9	14	15	12	13	18	19	16	17	22	23	20	21	26	27	24	25	30	31	28	29
3	-	-	7	6	5	4	11	10	9	8	15	14	13	12	19	18	17	16	23	22	21	20	27	26	25	24	31	30	29	28
4	-	-	-	1	2	3	12	13	14	15	8	9	10	11	20	21	22	23	16	17	18	19	28	29	30	31	24	25	26	27
5	-	-	-	-	3	2	13	12	15	14	9	8	11	10	21	20	23	22	17	16	19	18	29	28	31	30	25	24	27	26
6	-	-	-	-	-	1	14	15	12	13	10	11	8	9	22	23	20	21	18	19	16	17	30	31	28	29	26	27	24	25
7	-	-	-	-	-	-	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
8	-	-	-	-	-	-	-	1	2	3	4	5	6	7	24	25	26	27	28	29	30	31	16	17	18	19	20	21	22	23
9	-	-	-	-	-	-	-	-	3	2	5	4	7	6	25	24	27	26	29	28	31	30	17	16	19	18	21	20	23	22
10	-	-	-	-	-	-	-	-	-	1	6	7	4	5	26	27	24	25	30	31	28	29	18	19	16	17	22	23	20	21
11	-	-	-	-	-	-	-	-	-	-	7	6	5	4	27	26	25	24	31	30	29	28	19	18	17	16	23	22	21	20
12	-	-	-	-	-	-	-	-	-	-	-	1	2	3	28	29	30	31	24	25	26	27	20	21	22	23	16	17	18	19
13	-	-	-	-	-	-	-	-	-	-	-	-	3	2	29	28	31	30	25	24	27	26	21	20	23	22	17	16	19	18
14	-	-	-	-	-	-	-	-	-	-	-	-	-	1	30	31	28	29	26	27	24	25	22	23	20	21	18	19	16	17
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2	5	4	7	6	9	8	11	10	13	12	15	14
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	6	7	4	5	10	11	8	9	14	15	12	13
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	6	5	4	11	10	9	8	15	14	13	12
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3	12	13	14	15	8	9	10	11
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2	13	12	15	14	9	8	11	10
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	14	15	12	13	10	11	8	9
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15	14	13	12	11	10	9	8
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3	4	5	6	7
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2	5	4	7	6
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	6	7	4	5
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	6	5	4
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1

We can also use this notation to easily separate the set of WP factor candidates from the SP factor candidates. The WP factor candidate numbers are $1 \rightarrow 2^{n_1-k_1}-1$, with the remaining numbers, $(2^{n_1-k_1} \rightarrow 2^{(n_1+n_2)-(k_1+k_2)}-1)$, corresponding to the SP factor candidates. For a 32-run FFSP design with $n_1-k_1=4$ and $n_2-k_2=1$, the WP factor

candidates numbers are $1 \rightarrow 15$. That is, any letter combination with letters **A**, **B**, **C**, or **D**. The SP factor candidate numbers are $16 \rightarrow 31$. They correspond to letter combinations with letter **E**.

Now that we have a representation for the set of letter groups for the WP and SP factor candidates, let's consider the integer programming (IP) model to select the optimal factors.

Integer Programming Model

IP models are widely used in the operations field to model the system with an overarching objective function and a collection of constraints to restrict the solution space. There are several commercially available software titles to represent and solve IP models. We are using AMPL, an algebraic modeling language, to code the formulation of the system. Within, AMPL, we are using the CPLEX 10 solver engine to solve the IPs. CPLEX takes advantage of the latest techniques and options available for solve IPs, including various branch & bound and relaxation strategies. For the remainder of the examples in this paper, we will use the default CPLEX options within AMPL to find the optimal solution answer. All of the solutions provided take less than 15 seconds to solve for the feasible representations. The infeasible models are identified immediately within AMPL.

Suppose that in a $2^{(n_1+n_2)-(k_1+k_2)}$ FFSP design with $p_1 = 2^{(n_1-k_1)} - 1$ and $p_2 = 2^{(n_1+n_2)-(k_1+k_2)} - 2^{(n_1-k_1)}$ we would like to allocate n_1 WP factors into p_1 candidate columns and n_2 SP factors into p_2 candidate columns. We define a binary decision variable x_k which is equal to 1 if the k^{th} letter group is chosen. The variables x_k for

$k=1,\dots,p_1$ correspond to the WP factors and the variables x_k for $k=p_1,\dots,p_1+p_2$ correspond to the SP factors. The following constraints will give an appropriate FFSP.

$$\sum_{k=1}^{p_1} x_k = n_1 \tag{2.1.1}$$

$$\sum_{k=p_1+1}^{p_1+p_2} x_k = n_2 \tag{2.1.2}$$

Constraints (2.1.1) and (2.1.2) set the number of main effects, where the model selects n_1 number of WP factors and n_2 number of SP factors.

$$x_i + x_j - w_{ij} \le 1, i<j \tag{2.2.1}$$

$$x_i + x_j - 2w_{ij} \ge 0, i<j \tag{2.2.2}$$

Constraints (2.2.1) and (2.2.2) enforce that the binary variable $w_{ij}=1$ if and only if $x_i=1$ and $x_j=1$.

$$\sum_{i,j\in S_k} w_{ij} = y_k + z_k, k=1,\dots,p_1+p_2 \tag{2.3.1}$$

$$z_k \le M(1-y_k), k=1,\dots,p_1+p_2 \tag{2.3.2}$$

$$y_k + z_k \le M\tau_k, k=1,\dots,p_1+p_2 \tag{2.3.3}$$

where S_k is the set of 2FIs that are equivalent to the k^{th} letter group. Constraints (2.3.1) and (2.3.2) represent the number of 2FIs corresponding to each letter group. Here, we introduce the binary variable y_k, which is equal to 1 if exactly one 2FI equivalent to the k^{th} letter group is chosen. When there are more than one 2FI equivalent to the k^{th} letter group, we use the integer variable z_k to represent this number. Constraint (2.3.3) sets

binary variable $\tau_k = 1$ if there is at least one 2FI equivalent to the k^{th} letter group. This constraint also used the Big-M to reference a very large number. This is a common practice in IP modeling to turn on or off binary variables, and is used extensively throughout our IP model.

$$y_k + (1 - x_k) - t_k \le 1,\ k = 1,\ldots, p_1 + p_2 \tag{2.4.1}$$

$$y_k + (1 - x_k) - 2t_k \ge 0,\ k = 1,\ldots, p_1 + p_2 \tag{2.4.2}$$

$$\sum_{k=1}^{p_1+p_2} t_k = Clear2FIs \tag{2.4.3}$$

Constraints (2.4.1) – (2.4.3) are used to determine the number of clear 2FIs. The first two constraints set the binary variable t_k to 1 only when there is one 2FI corresponding to the k^{th} letter group ($y_k = 1$) and the main effect for that letter group is not selected ($x_k = 0$). The last constraint then sums over all letter groups to find the number of clear 2FIs.

The objective function below (2.5) maximizes the number of clear 2FIs

$$\text{Max } Z = Clear2FIs \tag{2.5}$$

There are times when the experimenter desires a Resolution IV design, where the main effects are clear from confounding with any 2FIs. In order to accomplish this resolution in our IP formulation, we add constraints (2.6). These constraints make sure that if the main effect $x_k = 1$ is selected then no 2FI can be selected.

$$\tau_k \le M(1 - x_k),\ k = 1,\ldots, p_1 + p_2 \tag{2.6}$$

Let's look at a small FFSP example to understand these constraints further. Suppose we wish to create a 32-run FFSP design with 4 WP factors and 5 SP factors using this IP

formulation. We will use 8 whole plots, each having 4 subplot runs. Thus, $n_1 - k_1 = 4 - 1 = 3$ and $n_2 - k_2 = 5 - 3 = 2$. The first two IP formulation constraints are

$$\sum_{1}^{7} x_k = 4$$
$$\sum_{8}^{31} x_k = 5$$

The next set of constraints determines which 2FI variables are turned on. So if we choose letter group **AC** ($x_5 = 1$) as a WP factor and **ABD** ($x_{11} = 1$) as a SP factor, then the first constraint mandates $w_{5,11} = 1$. Conversely, the second constraint says that if $w_{5,11} = 1$, then the other two main effects must be selected.

$$x_5 + x_{11} - w_{5,11} \leq 1$$
$$x_5 + x_{11} - 2w_{5,11} \geq 0$$

From Table 2, we know that (5,11) corresponds to letter group 14, or **BCD**. It can be shown that with an *n*-run design, there are (*n*-2)/2 2FIs corresponding to each of the (*n*-1) letter groups. Thus, for this 32-run design, we have 15 2FI variables for each of the 31 letter groups. Constraint (2.3.1), corresponding to letter group **BCD**, can be seen as an alias chain.

$$\begin{aligned}\sum_{i,j\in S_{14}} w_{ij} &= w_{1,15} + w_{2,12} + w_{3,13} + w_{4,10} + w_{5,11} + w_{6,8} + w_{7,9} + w_{16,30} + w_{17,31} \\ &\quad + w_{18,28} + w_{19,29} + w_{20,26} + w_{21,27} + w_{22,24} + w_{23,25} \\ &= y_{14} + z_{14}\end{aligned}$$

Let's assume our WP factors are [**A B C AC**] and the SP factors are [**D E CD ABD ABCDE**]. The corresponding numeric values are [**1 2 4 5**] and [**8 16 12 11 31**] respectively. Then for letter group 14, (2.3.1) becomes

$$\sum_{i,j \in S_{14}} w_{ij} = w_{2,12} + w_{5,11} = 1 + 1 = 2 = y_{14} + z_{14}$$

This corresponds to an alias chain with two 2FIs aliased with each other. Constraint (2.3.2) sets $z_{14} = 2$ and $y_{14} = 0$ since y is a binary variable, and constraint (2.3.3) sets $t_{14} = 1$ to represent that there is at least one 2FI corresponding to the 14th letter group.

Finally, for a Resolution IV design, there cannot be any 2FIs corresponding to letter group k if the corresponding main effect is selected, $x_k = 1$. Thus, constraint (2.6) restricts the design to Resolution IV or greater. For $k = 14$, (2.6) becomes

$$t_{14} \leq M(1 - x_{14})$$

Since $t_{14} = 2$, then **BCD** cannot be a design generator or main effect ($x_{14} = 0$).

For Resolution III designs, we simply need to relax the IP by removing constraint (2.6). This allows main effects to be aliased with 2FIs; however, 2FIs are not considered "clear" when aliased with a main effect, and therefore are not counted in the objective function.

Customized Constraints

We will now add addition constraints to the IP formulation to customize the design for isolating specific main effects and 2FIs. Let's say we must have one WP factor and all its 2FIs (both WP2FI and WS2FI) to be clear from confounding. We know that letter group $x_1 = 1$ refers to a WP factor that must be chosen since it represents a single letter, **A**. Thus, we add constraint (2.7) to require the main effect, x_1, to be clear of any 2FI. Notice, constraint (2.7) is the same as (2.6) for $k = 1$. We only need to add (2.7) when we are not using a Resolution IV IP formulation.

$$t_1 \leq M(1 - x_1) \tag{2.7}$$

Constraints (2.8.1) and (2.8.2) make sure all 2FIs involving this main effect are clear.

$$y_k + z_k + x_k \leq M(1 - w_{1,s}) + 1, s = 1, \ldots, p_1 + p_2 \tag{2.8.1}$$

$$y_k + z_k + x_k \leq M(1 - w_{s,1}) + 1, s = 1, \ldots, p_1 + p_2 \tag{2.8.2}$$

where k is the letter group from Table 2 corresponding to $w_{1,s}$ in the first constraint and $w_{s,1}$ in the second constraint. If we only want the WS2FIs for this WP main effect to be clear, we change the limitations on constraints (2.8.1) and (2.8.2) as follows:

$$y_k + z_k + x_k \leq M(1 - w_{1,s}) + 1, s = p_1 + 1, \ldots, p_1 + p_2 \tag{2.8.1}$$

$$y_k + z_k + x_k \leq M(1 - w_{s,1}) + 1, s = p_1 + 1, \ldots, p_1 + p_2 \tag{2.8.2}$$

Thus, adding constraints (2.7), (2.8.1) and (2.8.2) restrict the solution space for the IP model so that all feasible solutions meet the customized constraint. The objective function then acts as a secondary goal of maximizing the number of clear 2FIs.

Example 1: 16-run FFSP design

Suppose we have a case with 3 whole plot factors and 5 subplot factors. We will further assume only 2-level FFSP designs are considered. In order to obtain at least a Resolution III design, we must have more than 8 runs to estimate the main effects, so we will consider a 16-run design and look at 2 cases for various numbers of whole plots.

Case 1: 4 WPs and 4 SP runs for each WP.

This case allows for up to 3 WP factors and 12 SP factors in a fully saturated design. We require 3 WP factors and 5 SP factors, "saturating" the WP part of the design. This number of WPs also limits out design to Resolution III. This is confirmed by the IP

model, since having constraint (2.6) in the model results in an infeasible model formulation.

For a Resolution III design, we remove constraint (2.6) from the model. The resulting model produces a design with 7 clear 2FIs shown in Table 3.

Table 3: 16-run Res III FFSP with 4 WPs and 7 clear 2FIs

Standard order	Plot#	WP factors			SP factors				
		A	B	AB	C	D	AD	BD	ABD
1	1	-1	-1	1	-1	-1	1	1	-1
2	1	-1	-1	1	-1	1	-1	-1	1
3	1	-1	-1	1	1	-1	1	1	-1
4	1	-1	-1	1	1	1	-1	-1	1
5	2	-1	1	-1	-1	-1	1	-1	1
6	2	-1	1	-1	-1	1	-1	1	-1
7	2	-1	1	-1	1	-1	1	-1	1
8	2	-1	1	-1	1	1	-1	1	-1
9	3	1	-1	-1	-1	-1	-1	1	1
10	3	1	-1	-1	-1	1	1	-1	-1
11	3	1	-1	-1	1	-1	-1	1	1
12	3	1	-1	-1	1	1	1	-1	-1
13	4	1	1	1	-1	-1	-1	-1	-1
14	4	1	1	1	-1	1	1	1	1
15	4	1	1	1	1	-1	-1	-1	-1
16	4	1	1	1	1	1	1	1	1

All main effects are aliased with 3 2FIs, except for one SP factor, **C**, which is completely clear of any 2FIs. Also, the 7 clear two-factor interactions are those that involve subplot factor **C**. This design could be very useful when we are primarily concerned with the effect of one of the SP factors.

Note this design is "saturated" in the WP design, with effectively only 4 runs (4 whole plots) at the WP level. This allows for only 4 degrees of freedom to estimate the grand average and the effect estimates of the 3 WP factors. With all the degrees of freedom used, we are limited in testing the significance of the WP main effects.

Case 2: 8 WPs and 2 SP runs for each WP.

The second case for a 16-run FFSP design involves 8 WPs where there are only 2 SP runs for each WP. This corresponds to having $n_1 = 3$ and $k_1 = 0$ for the WP design and $n_2 = 5$ and $k_2 = 4$ for the SP design. From the set of 15 letter combinations, the first 7 represent candidates for the 3 WP factors and the remaining 8 are available for the 5 SP factors. This is represented in the IP model by adjusting constraints (2.1.1) and (2.1.2) as follows:

$$\sum_{1}^{7} x_k = 3 \tag{2.1.1}$$

$$\sum_{8}^{15} x_k = 5 \tag{2.1.2}$$

Since Resolution IV designs are preferable, we initially model this case with constraint (2.6). We quickly determine during the presolve that there is no feasible solution to this IP formulation; hence we have no Resolution IV design available in a 16-run FFSP design. Removing the Resolution IV constraint (2.6) from the provides us with the design in Table 4.

Table 4: 16-run Res III FFSP with 8 WPs and 2 clear 2FIs

Standard order	Plot #	Whole plot factors A	B	C	Subplot factors D	AD	BD	ABD	CD
1	1	-1	-1	-1	-1	1	1	-1	1
2	1	-1	-1	-1	1	-1	-1	1	-1
3	2	-1	-1	1	-1	1	1	-1	-1
4	2	-1	-1	1	1	-1	-1	1	1
5	3	-1	1	-1	-1	1	-1	1	1
6	3	-1	1	-1	1	-1	1	-1	-1
7	4	-1	1	1	-1	1	-1	1	-1
8	4	-1	1	1	1	-1	1	-1	1
9	5	1	-1	-1	-1	-1	1	1	1
10	5	1	-1	-1	1	1	-1	-1	-1
11	6	1	-1	1	-1	-1	1	1	-1
12	6	1	-1	1	1	1	-1	-1	1
13	7	1	1	-1	-1	-1	-1	-1	1
14	7	1	1	-1	1	1	1	1	-1
15	8	1	1	1	-1	-1	-1	-1	-1
16	8	1	1	1	1	1	1	1	1

None of the main effects are clear and there are only 2 clear 2FIs. Another alternative design is presented by Bingham and Sitter (2001), where they look at choosing where to split the SPD. In some cases, the system will force the experimenter to know which factors should be treated as WP factors and which should be treated as SP factors. In other systems, a SP factor can be treated as a WP factor, maintaining its level throughout each WP, such as the design shown in Table 5.

Table 5: 16-run Res III FFSP with 8 WPs treating one SP factor (ABC) as a WP factor. There are 7 clear 2FIs

Standard	Plot#	Whole plot factors			Subplot factors				
order		A	B	C	ABC	D	AD	BD	ABD
1	1	-1	-1	-1	-1	-1	1	1	-1
2	1	-1	-1	-1	-1	1	-1	-1	1
3	2	-1	-1	1	1	-1	1	1	-1
4	2	-1	-1	1	1	1	-1	-1	1
5	3	-1	1	-1	1	-1	1	-1	1
6	3	-1	1	-1	1	1	-1	1	-1
7	4	-1	1	1	-1	-1	1	-1	1
8	4	-1	1	1	-1	1	-1	1	-1
9	5	1	-1	-1	1	-1	-1	1	1
10	5	1	-1	-1	1	1	1	-1	-1
11	6	1	-1	1	-1	-1	-1	1	1
12	6	1	-1	1	-1	1	1	-1	-1
13	7	1	1	-1	-1	-1	-1	-1	-1
14	7	1	1	-1	-1	1	1	1	1
15	8	1	1	1	1	-1	-1	-1	-1
16	8	1	1	1	1	1	1	1	1

This design provides 7 clear 2FIs in addition to the 8 clear main effects, but we have less precision when estimating the SP main effect that was moved into the WP category since it is tested against the WP error. While this strategy can be used in some situations, we are assuming that the SP main effects in our systems must change throughout the whole plot. Considering SP effects as WP factors is beyond the scope of this paper. Therefore, due to the limited number of clear effects and the difficulty in changing "hard" to change factors, we recommend using the design found in Case 1. Moreover, since running WP's is usually more expensive, a possible alternative strategy is to replicate the design in Case 1. This will allow for testing the WP effects by only running the same amount of WP's as in Case 2. Further improvements can be obtained in Case 1 by adding follow-up runs as discussed in Almimi et al. (2008).

Example 2: 32-run FFSP design

From the above cases, we see that there is no Resolution IV FFSP design with only 16 runs for 3 SP factors and 5 SP factors. In addition, there is little flexibility in customizing the alias structure for the Resolution III design. In order to increase our design resolution and gain more customization in estimating effects, we will look at 3 cases from the 32-run FFSP design: 4, 8, and 16 whole plots.

Case 1: 4 WPs and 8 SP runs for each WP.

With only 4 WPs, there is only one choice for the WP factors, [**A B AB**], while the remaining 28 letter combinations are candidates for the 5 SP factors. Once again, this restriction at the whole plot level limits our design to Resolution III, so we will not even attempt to run the model with constraint (2.6). The IP formulation results in a Resolution III design with 18 clear two-factor interactions.

Suppose we wish to customize this design by trying to isolate the WP main effects and their 2FIs. Due to the number of WP runs, each WP factor must be aliased with a WP2FI, so we will focus on clear WS2FIs. We systematically add constraints (2.8.1) and (2.8.2) for each WP factor to require their WS2FIs to remain clear. The design shown in Table 6 allows for 15 clear 2FIs so that all WS2FIs for the 3 WP factors are clear.

Table 6: 32-run Res III FFSP with 4 WPs and 15 clear 2FIs. All WS2FIs for the three WP factors are clear.

Standard	Plot#	Whole plot factors			Subplot factors				
order		A	B	AB	C	D	E	DE	CDE
1	1	-1	-1	1	-1	-1	-1	1	-1
2	1	-1	-1	1	-1	-1	1	-1	1
3	1	-1	-1	1	-1	1	-1	-1	1
4	1	-1	-1	1	-1	1	1	1	-1
5	1	-1	-1	1	1	-1	-1	1	1
6	1	-1	-1	1	1	-1	1	-1	-1
7	1	-1	-1	1	1	1	-1	-1	-1
8	1	-1	-1	1	1	1	1	1	1
9	2	-1	1	-1	-1	-1	-1	1	-1
10	2	-1	1	-1	-1	-1	1	-1	1
11	2	-1	1	-1	-1	1	-1	-1	1
12	2	-1	1	-1	-1	1	1	1	-1
13	2	-1	1	-1	1	-1	-1	1	1
14	2	-1	1	-1	1	-1	1	-1	-1
15	2	-1	1	-1	1	1	-1	-1	-1
16	2	-1	1	-1	1	1	1	1	1
17	3	1	-1	-1	-1	-1	-1	1	-1
18	3	1	-1	-1	-1	-1	1	-1	1
19	3	1	-1	-1	-1	1	-1	-1	1
20	3	1	-1	-1	-1	1	1	1	-1
21	3	1	-1	-1	1	-1	-1	1	1
22	3	1	-1	-1	1	-1	1	-1	-1
23	3	1	-1	-1	1	1	-1	-1	-1
24	3	1	-1	-1	1	1	1	1	1
25	4	1	1	1	-1	-1	-1	1	-1
26	4	1	1	1	-1	-1	1	-1	1
27	4	1	1	1	-1	1	-1	-1	1
28	4	1	1	1	-1	1	1	1	-1
29	4	1	1	1	1	-1	-1	1	1
30	4	1	1	1	1	-1	1	-1	-1
31	4	1	1	1	1	1	-1	-1	-1
32	4	1	1	1	1	1	1	1	1

We should note again that having only four WPs with 3 WP factors leaves no test for significance among the WP effects.

Case 2: 8 WPs and 4 SP runs for each WP.

By changing the limits on the summations in constraints (2.1.1) and (2.1.2), we can increase the number of WPs to 8, so letter groups corresponding to numbers $1 \rightarrow 7$ are candidates for the WP factors and the remaining 24 letter groups are candidates for the

SP factors. Using the basic IP model with constraint (2.6) will find a Resolution IV design with the greatest number of clear 2FIs. This results in a design with 13 clear two-factor interactions. Next, we add constraints (2.7), (2.8.1) and (2.8.2) in a systematic manner to customize the design by distributing the clear 2FIs, while maintaining our design resolution. For this case, we can isolate all the 2FIs involving up to 2 of the WP factors using the design in Table 7.

Table 7: 32-run Res IV FFSP with 8 WPs and 13 clear 2FIs. All 2FIs for WP factors A and B are clear.

Standard order	Plot #	Whole plot factors A	B	C	Subplot factors D	ABCD	E	ABDE	CDE
1	1	-1	-1	-1	-1	1	-1	1	-1
2	1	-1	-1	-1	-1	1	1	-1	1
3	1	-1	-1	-1	1	-1	-1	-1	1
4	1	-1	-1	-1	1	-1	1	1	-1
5	2	-1	-1	1	-1	-1	-1	1	1
6	2	-1	-1	1	-1	-1	1	-1	-1
7	2	-1	-1	1	1	1	-1	-1	-1
8	2	-1	-1	1	1	1	1	1	1
9	3	-1	1	-1	-1	-1	-1	-1	-1
10	3	-1	1	-1	-1	-1	1	1	1
11	3	-1	1	-1	1	1	-1	1	1
12	3	-1	1	-1	1	1	1	-1	-1
13	4	-1	1	1	-1	1	-1	-1	1
14	4	-1	1	1	-1	1	1	1	-1
15	4	-1	1	1	1	-1	-1	1	-1
16	4	-1	1	1	1	-1	1	-1	1
17	5	1	-1	-1	-1	-1	-1	-1	-1
18	5	1	-1	-1	-1	-1	1	1	1
19	5	1	-1	-1	1	1	-1	1	1
20	5	1	-1	-1	1	1	1	-1	-1
21	6	1	-1	1	-1	1	-1	-1	1
22	6	1	-1	1	-1	1	1	1	-1
23	6	1	-1	1	1	-1	-1	1	-1
24	6	1	-1	1	1	-1	1	-1	1
25	7	1	1	-1	-1	1	-1	1	-1
26	7	1	1	-1	-1	1	1	-1	1
27	7	1	1	-1	1	-1	-1	-1	1
28	7	1	1	-1	1	-1	1	1	-1
29	8	1	1	1	-1	-1	-1	1	1
30	8	1	1	1	-1	-1	1	-1	-1
31	8	1	1	1	1	1	-1	-1	-1
32	8	1	1	1	1	1	1	1	1

Removing constraint (2.6) results in a Resolution III design with 18 clear 2FIs. Similar to the Resolution IV case, we can add the customized constraints to this IP model to find the design in Table 8 where all three whole plot factors are clear along with their WS2FIs.

Table 8: 32-run Res III FFSP with 8 WPs and 18 clear 2FIs. All WS2FIs are clear from confounding.

Standard	Plot #	Whole plot factors			Subplot factors				
order		A	B	C	D	ABCD	E	ABCE	DE
1	1	-1	-1	-1	-1	1	-1	1	1
2	1	-1	-1	-1	-1	1	1	-1	-1
3	1	-1	-1	-1	1	-1	-1	1	-1
4	1	-1	-1	-1	1	-1	1	-1	1
5	2	-1	-1	1	-1	-1	-1	-1	1
6	2	-1	-1	1	-1	-1	1	1	-1
7	2	-1	-1	1	1	1	-1	-1	-1
8	2	-1	-1	1	1	1	1	1	1
9	3	-1	1	-1	-1	-1	-1	-1	1
10	3	-1	1	-1	-1	-1	1	1	-1
11	3	-1	1	-1	1	1	-1	-1	-1
12	3	-1	1	-1	1	1	1	1	1
13	4	-1	1	1	-1	1	-1	1	1
14	4	-1	1	1	-1	1	1	-1	-1
15	4	-1	1	1	1	-1	-1	1	-1
16	4	-1	1	1	1	-1	1	-1	1
17	5	1	-1	-1	-1	-1	-1	-1	1
18	5	1	-1	-1	-1	-1	1	1	-1
19	5	1	-1	-1	1	1	-1	-1	-1
20	5	1	-1	-1	1	1	1	1	1
21	6	1	-1	1	-1	1	-1	1	1
22	6	1	-1	1	-1	1	1	-1	-1
23	6	1	-1	1	1	-1	-1	1	-1
24	6	1	-1	1	1	-1	1	-1	1
25	7	1	1	-1	-1	1	-1	1	1
26	7	1	1	-1	-1	1	1	-1	-1
27	7	1	1	-1	1	-1	-1	1	-1
28	7	1	1	-1	1	-1	1	-1	1
29	8	1	1	1	-1	-1	-1	-1	1
30	8	1	1	1	-1	-1	1	1	-1
31	8	1	1	1	1	1	-1	-1	-1
32	8	1	1	1	1	1	1	1	1

The customized constraints can also be used to isolate SP factors and their interactions. If we add constraints (2.7), (2.8.1), and (2.8.2) for factors [**A B D E**], we find the design in Table 9 where the 2 WP factors **A** and **B** and their WS2FIs are clear. In addition, the 2 SP factors **D** and **E** are clear along with their WS2FIs with all WP factors.

Table 9: 32-run Res III FFSP with 8 WPs and 18 clear 2FIs. All WS2FIs for WP factors A and B and SP factors D and E are clear from confounding.

Standard order	Plot #	Whole plot factors			Subplot factors				
		A	B	AC	D	BCD	E	BCE	BCDE
1	1	-1	-1	1	-1	-1	-1	-1	1
2	1	-1	-1	1	-1	-1	1	1	-1
3	1	-1	-1	1	1	1	-1	-1	-1
4	1	-1	-1	1	1	1	1	1	1
5	2	-1	-1	-1	-1	1	-1	1	-1
6	2	-1	-1	-1	-1	1	1	-1	1
7	2	-1	-1	-1	1	-1	-1	1	1
8	2	-1	-1	-1	1	-1	1	-1	-1
9	3	-1	1	1	-1	1	-1	1	-1
10	3	-1	1	1	-1	1	1	-1	1
11	3	-1	1	1	1	-1	-1	1	1
12	3	-1	1	1	1	-1	1	-1	-1
13	4	-1	1	-1	-1	-1	-1	-1	1
14	4	-1	1	-1	-1	-1	1	1	-1
15	4	-1	1	-1	1	1	-1	-1	-1
16	4	-1	1	-1	1	1	1	1	1
17	5	1	-1	1	-1	1	-1	1	-1
18	5	1	-1	1	-1	1	1	-1	1
19	5	1	-1	1	1	-1	-1	1	1
20	5	1	-1	1	1	-1	1	-1	-1
21	6	1	-1	-1	-1	-1	-1	-1	1
22	6	1	-1	-1	-1	-1	1	1	-1
23	6	1	-1	-1	1	1	-1	-1	-1
24	6	1	-1	-1	1	1	1	1	1
25	7	1	1	1	-1	-1	-1	-1	1
26	7	1	1	1	-1	-1	1	1	-1
27	7	1	1	1	1	1	-1	-1	-1
28	7	1	1	1	1	1	1	1	1
29	8	1	1	-1	-1	1	-1	1	-1
30	8	1	1	-1	-1	1	1	-1	1
31	8	1	1	-1	1	-1	-1	1	1
32	8	1	1	-1	1	-1	1	-1	-1

Case 3: 16 WPs and 2 SP runs for each WP.

The final case for a 32-run FFSP design involves 16 WP's with $n_1 = 4$, $k_1 = 0$, $n_2 = 5$ and $k_2 = 4$. However, since we only have 3 WP factors, we can only distinguish between 8 different plots, so with 16 plots we are actually running two replicates of an 8

plot design at the WP design level. We modify constraints (2.1.1) and (2.1.2) to partition the candidate letter combinations as follows:

$$\sum_{1}^{15} x_k = 3 \tag{2.1.1}$$

$$\sum_{16}^{31} x_k = 5 \tag{2.1.2}$$

Running the IP model for a Resolution IV design yields a design with 13 clear 2FIs. In an attempt to customize this design using constraints (2.8.1) and (2.8.2), we find that it is impossible to isolate any WP factors and their WS2FIs. However, we are able to add these constraints and isolate SP factor E and its WS2FIs in following design.

Table 10: 32-run Res IV FFSP with 2 replicates of the 8 WPs and 13 clear 2FIs. SP factor E and its entire cross 2FIs are clear from confounding.

Standard order	Plot #	WP factors			SP factors				
		A	B	C	E	ABCE	ADE	CDE	ABCDE
1	1	-1	-1	-1	-1	1	-1	-1	-1
2	1	-1	-1	-1	1	-1	1	1	1
3	2	-1	-1	1	-1	-1	-1	1	1
4	2	-1	-1	1	1	1	1	-1	-1
5	3	-1	1	-1	-1	-1	-1	-1	1
6	3	-1	1	-1	1	1	1	1	-1
7	4	-1	1	1	-1	1	-1	1	-1
8	4	-1	1	1	1	-1	1	-1	1
9	5	1	-1	-1	-1	-1	1	-1	1
10	5	1	-1	-1	1	1	-1	1	-1
11	6	1	-1	1	-1	1	1	1	-1
12	6	1	-1	1	1	-1	-1	-1	1
13	7	1	1	-1	-1	1	1	-1	-1
14	7	1	1	-1	1	-1	-1	1	1
15	8	1	1	1	-1	-1	1	1	1
16	8	1	1	1	1	1	-1	-1	-1
17	9	-1	-1	-1	-1	1	1	1	1
18	9	-1	-1	-1	1	-1	-1	-1	-1
19	10	-1	-1	1	-1	-1	1	-1	-1
20	10	-1	-1	1	1	1	-1	1	1
21	11	-1	1	-1	-1	-1	1	1	-1
22	11	-1	1	-1	1	1	-1	-1	1
23	12	-1	1	1	-1	1	1	-1	1
24	12	-1	1	1	1	-1	-1	1	-1
25	13	1	-1	-1	-1	-1	-1	1	-1
26	13	1	-1	-1	1	1	1	-1	1
27	14	1	-1	1	-1	1	-1	-1	1
28	14	1	-1	1	1	-1	1	1	-1
29	15	1	1	-1	-1	1	-1	1	1
30	15	1	1	-1	1	-1	1	-1	-1
31	16	1	1	1	-1	-1	-1	-1	-1
32	16	1	1	1	1	1	1	1	1

For a Resolution III design, we are able to find a design with 18 clear 2FIs. Adding the customized constraints allows us to find the design in Table 11 with WP factor **A** and SP factor **E** clear, along with their WS2FIs.

Table 11: 32-run Res III FFSP with 2 replicates of the 8 WPs and 18 clear 2FIs. WP factor A and SP factor E are clear along with their cross 2FIs.

Standard	Plot #	WP factors			SP factors				
order		A	B	C	E	ABCE	ADE	ABDE	ACDE
1	1	-1	-1	-1	-1	1	-1	1	1
2	1	-1	-1	-1	1	-1	1	-1	-1
3	2	-1	-1	1	-1	-1	-1	1	-1
4	2	-1	-1	1	1	1	1	-1	1
5	3	-1	1	-1	-1	-1	-1	-1	1
6	3	-1	1	-1	1	1	1	1	-1
7	4	-1	1	1	-1	1	-1	-1	-1
8	4	-1	1	1	1	-1	1	1	1
9	5	1	-1	-1	-1	-1	1	-1	-1
10	5	1	-1	-1	1	1	-1	1	1
11	6	1	-1	1	-1	1	1	-1	1
12	6	1	-1	1	1	-1	-1	1	-1
13	7	1	1	-1	-1	1	1	1	-1
14	7	1	1	-1	1	-1	-1	-1	1
15	8	1	1	1	-1	-1	1	1	1
16	8	1	1	1	1	1	-1	-1	-1
17	9	-1	-1	-1	-1	1	1	-1	-1
18	9	-1	-1	-1	1	-1	-1	1	1
19	10	-1	-1	1	-1	-1	1	-1	1
20	10	-1	-1	1	1	1	-1	1	-1
21	11	-1	1	-1	-1	-1	1	1	-1
22	11	-1	1	-1	1	1	-1	-1	1
23	12	-1	1	1	-1	1	1	1	1
24	12	-1	1	1	1	-1	-1	-1	-1
25	13	1	-1	-1	-1	-1	-1	1	1
26	13	1	-1	-1	1	1	1	-1	-1
27	14	1	-1	1	-1	1	-1	1	-1
28	14	1	-1	1	1	-1	1	-1	1
29	15	1	1	-1	-1	1	-1	-1	1
30	15	1	1	-1	1	-1	1	1	-1
31	16	1	1	1	-1	-1	-1	-1	-1
32	16	1	1	1	1	1	1	1	1

Both of these designs will provide more accuracy in estimating the effects of the WP factors due to the replication at the WP level. However; there's a huge trade-off for this accuracy advantage. First, we have to change the WP factors 16 times, which defeats the purpose of running the experiments in a split plot mode to avoid changing the hard-to-change factors frequently. Second, neither the Resolution III nor the Resolution IV design for this case provides a lot of flexibility in the alias structure. We would recommend running 8 WPs in a 32-run design. Not only does this allow you to test for significance on the main effects, but there's enough flexibility to isolate several of the main effects and their 2FIs.

Example 3: 64-run, 2^{9-3} FFSP design

Suppose an experimenter wants to perform a 64-run, 2^{9-3} design. Also, suppose this experimenter wants to estimate all 2FIs of six of these nine factors. The minimum MA FF design from Chen, Sun and Wu (1993) will not allow for the estimation of all 2FIs of any six factors, but Bingham and Sitter (1999) use a MA FFSP design to isolate all six factors into the SP with their 2FIs clear of other main effects and other 2FIs. This satisfies the original experimenter's goal for the effect estimates of the 2^{9-3} design; however, what if the experimenter did not want all the clear 2FIs to involve only the SP factors, but instead 4 WP factors and 2 SP factors. Let's set up our IP formulation on a FFSP design with $n_1 = 5$ and $k_1 = 1$ for the 16-run WP part of the design and $n_2 = 4$ and $k_2 = 2$ for the 4-run SP design within each whole plot. Applying constraints (2.7), (2.8.1) and (2.8.2) to the four whole-plot factors [**A B C D**] and the two subplot factors [**E F**] sets up an appropriate IP formulation to meet this goal. We are also using

constraint (2.6) to maintain a Resolution IV design. The resulting IP model yields the feasible solution shown in Table 12, with a total of 30 clear two-factor interactions.

Table 12: 64-run Res IV FFSP with 16 WPs and 30 clear 2FIs. Factors [A B C D E F] are all clear along with their 2FIs.

Standard order	Plot #	Whole plot factors					Subplot factors			
		A	B	C	D	ACD	E	F	ABDE	BCDF
1	1	-1	-1	-1	-1	-1	-1	-1	1	1
2	1	-1	-1	-1	-1	-1	-1	1	1	-1
3	1	-1	-1	-1	-1	-1	1	-1	-1	1
4	1	-1	-1	-1	-1	-1	1	1	-1	-1
5	2	-1	-1	-1	1	1	-1	-1	-1	-1
6	2	-1	-1	-1	1	1	-1	1	-1	1
7	2	-1	-1	-1	1	1	1	-1	1	-1
8	2	-1	-1	-1	1	1	1	1	1	1
9	3	-1	-1	1	-1	1	-1	-1	1	-1
10	3	-1	-1	1	-1	1	-1	1	1	1
11	3	-1	-1	1	-1	1	1	-1	-1	-1
12	3	-1	-1	1	-1	1	1	1	-1	1
13	4	-1	-1	1	1	-1	-1	-1	-1	1
14	4	-1	-1	1	1	-1	-1	1	-1	-1
15	4	-1	-1	1	1	-1	1	-1	1	1
16	4	-1	-1	1	1	-1	1	1	1	-1
17	5	-1	1	-1	-1	-1	-1	-1	-1	-1
18	5	-1	1	-1	-1	-1	-1	1	-1	1
19	5	-1	1	-1	-1	-1	1	-1	1	-1
20	5	-1	1	-1	-1	-1	1	1	1	1
21	6	-1	1	-1	1	1	-1	-1	1	1
22	6	-1	1	-1	1	1	-1	1	1	-1
23	6	-1	1	-1	1	1	1	-1	-1	1
24	6	-1	1	-1	1	1	1	1	-1	-1
25	7	-1	1	1	-1	1	-1	-1	-1	1
26	7	-1	1	1	-1	1	-1	1	-1	-1
27	7	-1	1	1	-1	1	1	-1	1	1
28	7	-1	1	1	-1	1	1	1	1	-1
29	8	-1	1	1	1	-1	-1	-1	1	-1
30	8	-1	1	1	1	-1	-1	1	1	1
31	8	-1	1	1	1	-1	1	-1	-1	-1
32	8	-1	1	1	1	-1	1	1	-1	1
33	9	1	-1	-1	-1	1	-1	-1	-1	1
34	9	1	-1	-1	-1	1	-1	1	-1	-1
35	9	1	-1	-1	-1	1	1	-1	1	1
36	9	1	-1	-1	-1	1	1	1	1	-1
37	10	1	-1	-1	1	-1	-1	-1	1	-1
38	10	1	-1	-1	1	-1	-1	1	1	1
39	10	1	-1	-1	1	-1	1	-1	-1	-1
40	10	1	-1	-1	1	-1	1	1	-1	1
41	11	1	-1	1	-1	-1	-1	-1	-1	-1
42	11	1	-1	1	-1	-1	-1	1	-1	1
43	11	1	-1	1	-1	-1	1	-1	1	-1
44	11	1	-1	1	-1	-1	1	1	1	1
45	12	1	-1	1	1	1	-1	-1	1	1
46	12	1	-1	1	1	1	-1	1	1	-1
47	12	1	-1	1	1	1	1	-1	-1	1
48	12	1	-1	1	1	1	1	1	-1	-1
49	13	1	1	-1	-1	1	-1	-1	1	-1
50	13	1	1	-1	-1	1	-1	1	1	1
51	13	1	1	-1	-1	1	1	-1	-1	-1
52	13	1	1	-1	-1	1	1	1	-1	1
53	14	1	1	-1	1	-1	-1	-1	-1	1
54	14	1	1	-1	1	-1	-1	1	-1	-1
55	14	1	1	-1	1	-1	1	-1	1	1
56	14	1	1	-1	1	-1	1	1	1	-1
57	15	1	1	1	-1	-1	-1	-1	1	1
58	15	1	1	1	-1	-1	-1	1	1	-1
59	15	1	1	1	-1	-1	1	-1	-1	1
60	15	1	1	1	-1	-1	1	1	-1	-1
61	16	1	1	1	1	1	-1	-1	-1	-1
62	16	1	1	1	1	1	-1	1	-1	1
63	16	1	1	1	1	1	1	-1	1	-1
64	16	1	1	1	1	1	1	1	1	1

Example 4: Comparison with computer-generated optimal designs

We now consider one of our previous examples to compare it with the design we would obtain from a widely-used statistical DOE software package, JMP 6.0. We will focus on the 32-run FFSP from Example 2 – Case 1, where there are 4 WPs in the design. JMP allows the user to select the number of runs, plots, and distinguish between hard-to-change (WP) and easy-to-change (SP) factors. The *D*-optimal design matrix for the scenario from Example 2 – Case 1 provided by JMP is shown in Table 13 (sorted by factors [A B P Q R] respectively).

Table 13: *D*-Optimal design provided by JMP for 32-run FFSP estimating all WS2FIs for WP factors.

Standard order	Plot#	Whole plot factors			Subplot factors				
		A	B	C	P	Q	R	S	T
1	1	-1	-1	-1	-1	-1	-1	1	-1
2	1	-1	-1	-1	-1	-1	1	-1	1
3	1	-1	-1	-1	-1	1	-1	1	1
4	1	-1	-1	-1	-1	1	1	-1	-1
5	1	-1	-1	-1	1	-1	-1	-1	1
6	1	-1	-1	-1	1	-1	1	1	-1
7	1	-1	-1	-1	1	1	-1	-1	-1
8	1	-1	-1	-1	1	1	1	1	1
9	2	-1	1	1	-1	-1	-1	-1	1
10	2	-1	1	1	-1	-1	1	1	-1
11	2	-1	1	1	-1	1	-1	1	1
12	2	-1	1	1	-1	1	1	-1	-1
13	2	-1	1	1	1	-1	-1	-1	-1
14	2	-1	1	1	1	-1	1	1	1
15	2	-1	1	1	1	1	-1	1	-1
16	2	-1	1	1	1	1	1	-1	1
17	3	1	-1	1	-1	-1	-1	1	1
18	3	1	-1	1	-1	-1	1	-1	-1
19	3	1	-1	1	-1	1	-1	-1	1
20	3	1	-1	1	-1	1	1	1	-1
21	3	1	-1	1	1	-1	-1	-1	-1
22	3	1	-1	1	1	-1	1	1	1
23	3	1	-1	1	1	1	-1	1	-1
24	3	1	-1	1	1	1	1	-1	1
25	4	1	1	-1	-1	-1	-1	-1	1
26	4	1	1	-1	-1	-1	1	1	1
27	4	1	1	-1	-1	1	-1	1	-1
28	4	1	1	-1	-1	1	1	-1	-1
29	4	1	1	-1	1	-1	-1	1	-1
30	4	1	1	-1	1	-1	1	-1	-1
31	4	1	1	-1	1	1	-1	-1	1
32	4	1	1	-1	1	1	1	1	1

This design matrix looks very similar to the one in Table 6 generated using our IP formulation. Both designs are for a 32-run Resolution III FFSP design with 4 WPs having 3 WP factors and 5 SP factors. Both designs also provide enough degrees of freedom to estimate the WS2FIs between WP and SP factors. Since we are using the IP formulation to model a first-order FFSP design which is orthogonally blocked, then our design matrix is also *D*-optimal when modeling main effects and unconfounded

interaction effects. However, the big difference lies in the simplicity of our alias structure, with our approach leading only to regular designs while JMP does not have this restriction. Table 14 shows the alias structure using the design using our IP formulation from Table 6. This structure is regular, meaning all the coefficients are either 1 or -1, and our customization of constraints allowed us to ensure the 15 2FIs between WP and SP factors were clear of other main effects or 2FIs.

Table 14: Alias structure for 32-run Res III FFSP design with 4 WPs provided by IP model. 15 clear 2FIs including all cross 2FIs between WP and SP factors.

Main effects aliased with 2FI within own plot structure
A = BC
B = AC
C = AB
P = ST
Q = RS
R = QS
S = PT + QR
T = PS
15 clear 2FIs involving WP factors and SP factors
AP, AQ, AR, AS, AT, BP, BQ, BR, BS, BT, CP, CQ, CR, CS, CT
Remaining 2FIs
PQ = RT
PR = QT

On the other hand, Table 15 shows the irregular alias structure provided by the FFSP design generated by the JMP software. Not only does it have fractional coefficients in the alias structure, but the JMP package does not allow the user to ensure which effects are completely clear from confounding. The user can only specify which effects are estimatable.

Table 15: Alias structure for 32-run Res III FFSP design with 4 WPs provided by JMP software. No clear 2FIs.

Main Effects
A = - BC
B = - AC
C = - AB
P = 0.25 * QT + 0.25 * RS + 0.5 * RT
Q = 0.25 * PT - 0.25 * RS
R = 0.25 * PS + 0.5 * PT - 0.25 * QS + 0.25 * ST
S = 0.25 * PR - 0.25 * QR + 0.25 * RT
T = 0.25 * PQ + 0.5 * PR + 0.25 * RS
Remaining 2FIs
AP = 0.25 * QT - 0.25 * RS
AQ = 0.25 * PT + 0.25 * RS - 0.5 * ST
AR = - 0.25 * PS + 0.25 * QS + 0.25 * ST
AS = - 0.25 * PR + 0.25 * QR - 0.5 * QT + 0.25 * RT
AT = 0.25 * PQ - 0.5 * QS + 0.25 * RS
BP = 0.25 * QT - 0.25 * RS
BQ = 0.25 * PT - 0.25 * RS
BR = - 0.25 * PS - 0.25 * QS + 0.25 * ST
BS = - 0.25 * PR - 0.25 * QR + 0.25 * RT
BT = 0.25 * PQ + 0.25 * RS
CP = - 0.25 * QT - 0.25 * RS + 0.5 * RT
CQ = - 0.25 * PT - 0.25 * RS - 0.5 * ST
CR = - 0.25 * PS + 0.5 * PT - 0.25 * QS - 0.25 * ST
CS = - 0.25 * PR - 0.25 * QR - 0.5 * QT - 0.25 * RT
CT = - 0.25 * PQ + 0.5 * PR - 0.5 * QS - 0.25 * RS

This same structure difference occurs for all the example cases we have shown in this paper. JMP does not have the option to isolate clear effects or 2FI, whereas our IP model does have this functionality. Also, JMP returns irregular *D*-optimal designs, but our IP model always returns regular *D*-optimal designs for feasible solutions. For all of the example cases, our designs were generated using AMPL 9.1 with CPLEX solver engine and took less than 15 seconds to solve for the feasible designs. For infeasible design formulations, such as Resolution IV designs with only 4 whole plots, AMPL would immediately return an infeasible error message.

Conclusion

We have discussed a new approach to representing and designing FFSP simply by using the binary representation of their letter notation instead of referencing the design

matrix. This eliminates the need to calculate the alias matrix using complex matrix multiplication and the inverse function. By formulating the problem using Integer Programming, we can immediately determine if our custom requirements are feasible. In addition, due to the nature of IP formulation, the more constraints we put on the model, the easier it is to find the optimal solution, since we are further shrinking the solution space.

CHAPTER 3: GENERATING BLOCKED FRACTIONAL FACTORIAL DESIGNS USING INTEGER PROGRAMMING

Introduction

Two-level fractional factorial (FF) designs are very useful for screening experiments (Box & Hunter, 1961); however, limitations on changing some factors can make the experiment very expensive, time-consuming, or even impossible to perform in a completely random order. Thus, a randomization restriction can be imposed on the design to create a split-plot structure. FF split-plot (FFSP) designs use FF designs for the whole plot/subplot structures to accommodate the randomization restriction and are orthogonal (Kempthorne, 1998). Originating from the agricultural field, the hard-to-change factors, or large areas of land, comprised the whole plots (WPs), while subplots (SPs) of land within these large areas were considered easy-to-change (Yates, 1935).

When all the runs cannot be performed under the same conditions, the FFSP designs are split into blocks. For example, time constraints on running each experimental unit may limit the number of units that can be processed in a single day of work. There may be unintended variations in the process when the system is shut down at the end of the day or prepared for operation at the beginning of the next day for experiments. In order to account for variability between days, each day is considered a block. Blocking can also be used on process material when there is not enough of the same batch to proceed with all the experimental runs. There has been several publications involving blocking 2-

level FF designs (Chen, Sun, & Wu, 1993; Kulahci, 2007; Zhang & Park, 2000), but very little attention has been paid to blocking FFSP designs.

Several authors have already made tables of FFSP designs when using common design criteria. Bingham and Sitter (1999; 2001) presented tables for 8-, 16-, and 32-run FFSP designs using a Minimum Aberration (MA) design criterion. The MA design criterion (Fries & Hunter, 1980) provides a way to distinguish between designs of maximum resolution. Another criterion is the maximum number of clear 2FIs. Wu and Wu (2002) refer to these designs as MaxC2 designs and discuss the rules regarding this criterion for fractional factorials. Although not limited to balanced FFSP designs, Goos and Vandebroek (2001; 2004) have studied optimal split-plot designs using *D*-optimality. Goos (2002) proved that split-plot designs where the levels of the sub-plot factors sum to zero within each whole plot are *D*-optimal. Jones and Goos (2007) describe an algorithm for finding tailor-made *D*-optimal FFSP designs that can handle flexible choices of sample size, both continuous and categorical factors, and may include interaction terms of any order. Kulahci et al. (2006) present a compelling argument for custom FFSP designs not simply based on a single criterion, but based on the alias structure, including estimating certain types of clear two-factor interactions.

In a previous publication (Capehart, Keha, Kulahci, & Montgomery, 2008), we provide an integer programming (IP) model to generate optimal FFSP designs for user-defined criteria based on clear main effects and two-factor interactions (2FIs). In this paper, we extend the previous work to blocked designs.

Blocking FFSP Designs

McLeod and Brewster (2004) look at blocking MA FFSP designs to provide "good" BFFSP designs for estimating WP and SP main effects and their 2FIs. They break down the blocking effects for these designs into three categories. Pure WP blocking involves generating blocking variables using only WP factors. Separation blocking uses blocking generators consisting of SP factors alone or a mix of WP and SP factors. Finally, mixed blocking involves both pure WP blocking and separation, where some of the blocking variables are pure WP, and separation, where other blocking variables include some SP factors. They provide a table for 32-run MA BFFSP designs based on this approach. In order to further distinguish between designs with the same word length pattern (WLP) and provide more information on the estimation capabilities of the various designs, they use following secondary optimality criteria:

1. The number of clear main effects
2. The number of clear 2FIs
3. The number of clear SP main effects
4. The number of clear SP 2FIs
5. The number of clear SP main effects tested against WP error
6. The number of clear SP 2FIs tested against WP error

Criteria (5) and (6) look at which error term should be used to determine the significance of the specified contrast. When analyzing split-plot designs, the significance of the factors is determined by comparing the WP treatment sum of squares (SS) to the WP error term and the SP treatment SS to the respective SP error term. Sometimes the alias

structure may cause confusion in determining which error term to use. Bisgaard (2000) discusses this when analyzing split-plot designs and his results are summarized by the following three rules (Bingham & Sitter, 2001):

1. WP main effects and interactions involving only WP factors are compared to the WP error.
2. SP main effects or interactions that are aliased with WP main effects or interactions involving only WP factors are compared to the WP error.
3. SP main effects and interactions involving at least one SP factor that are not aliased with WP main effects or interactions involving only WP factors are compared to the SP error.

McLeod and Brewster (McLeod & Brewster, 2006) follow their work on screening BFFSP designs by looking at blocked designs for Robust Parameter Designs (RPDs). RPDs involve two types of factors: control and noise. Control factors can be controlled during the production process, while the noise factors represent changes to the production process that cannot be controlled at the time of production. In order to achieve products that are robust to the variability in the noise factors, the experimenter is interested in designing the experiment to estimate the control effects and control-by-noise interactions.

Goos and Vandebroek (2001) point out that split-plot designs cause a loss in precision in estimating the whole plot coefficients, while increasing the precision in estimation of the sub-plot coefficients and the whole plot by sub-plot interactions. Box and Jones (1992) show that the error variance for the whole-plot and subplot surround that of the error variance for a completely randomized design (CRD), where

$\sigma^2_{sub} < \sigma^2_{CRD} < \sigma^2_{whole}$. Similar research showing the increased precision for the subplot treatments is found in Kulahci et al. (2006). There is little interest in estimating the noise effects for RPDs, since they cannot be controlled, so most researchers choose to run an RPD experiment as a split-plot design, with the control factors at the SP level and noise factors at the WP level to increase the precision for the effects we are interested in.

As with their other publication (McLeod & Brewster, 2004), these authors provide tables of optimal BFFSP designs first based on MA criteria. The secondary criteria are slightly different from the 2004 paper:

1. Number of clear control main effects
2. Number of clear control-by-noise 2FIs
3. Number of clear control-by-control 2FIs
4. Number of clear control main effects tested against WP error
5. Number of clear control-by-noise 2FIs tested against WP error
6. Number of clear control-by-control 2FIs tested against WP error

Yang et. al. (2006) provide Theorems and Lemmas discussing the parameters of BFFSP designs that will have clear main effects and/or 2FIs. They define 2FIs into three categories: whole plot-by-whole plot (WP2FI), subplot-by-subplot (SP2FI), and whole plot-by-subplot (WS2FI), but they do not specify the number of clear 2FIs or specifically which factors are involved in designs with clear 2FIs. Nor do their Theorems or Lemmas discuss the number of clear main effects. McLeod (2007) provides a catalog of optimal block sequence for 32-run MA BFFSP designs; however, the sequencing of the blocks is beyond the scope of our research.

The following assumptions are common in the generation of most BFFSP designs (Jacroux, 2006):

- Hierarchical ordering principle: lower order treatment effects are more likely to be significant than higher order treatment effects and effects of the same order are equally likely to be significant.
- Three-factor and higher order treatment interactions are negligible.
- An interaction between block effects is of equal importance to a block main effect.
- Treatment-by-block interactions are negligible.
- Block effects are more likely to be significant than treatment effects.

The Chrome-Plating Experiment

In this section, we provide a brief overview of a case study McLeod and Brewster considered when creating BFFSP designs for screening purposes and from an RPD perspective. An aerospace company wants to identify the factors affecting excessive pitting and cracking, as well as bad adhesion and smoothness of chrome across one of the parts in its chrome-plating process. Six factors were identified for this experiment: A, chrome concentration; B, chrome to sulfate ratio; C, bath temperature; p, etching current density; q, plating current density; and r, part geometry. Each factor has two levels. Three of the factors, A, B, and C, were considered WP factors since they were characteristics of the bath where the plating occurred and could only be changed once per day. The other three factors, p, q, and r, were easy to change and therefore considered SP factors. Since there was a restriction in the randomization of the factors, due to the

difficulty in changing the WP factors, the company chose to run the experiment as split-plot design.

Two parts could be plated each day in the bath, and there was 16 days available for the experiment, resulting in 32 parts being plated. Instead of using a full factorial design ($2^6 = 64$), this was a half fractional factorial design. The company also chose to divide the 16 experiment days into four 4-day weeks, and consider each week as a block. All of these constraints resulted in a BFFSP design with a 4:4:2 structure. The *structure* for a BFFSP design consists of three numbers: number of blocks, numbers of WPs per block, and number of SPs per WP. The aerospace company was running a design with 4 blocks, 4 WPs per block, and 2 SPs per WP in their chrome-plating experiment.

Model Representation

Consider a Fractional Factorial (FF) design with 2^{n-k} runs. There are n total factors in the design, along with k fractional generators. In such designs, the first $n-k$ factors are considered *basic factors.* The *basic factors* can be represented by single letters (Franklin & Bailey, 1977), while the remaining generators are formed using the interaction of these single letter factors. There exists a set of $2^{n-k}-1$ letter groups formed using the letter group notation, which can be arranged using Yates order as follows:

$$[\mathrm{A} \quad \mathrm{B} \quad \mathrm{AB} \quad \ldots \quad \mathrm{D} \quad \mathrm{AD} \quad \mathrm{BD} \quad \mathrm{ABD} \quad \ldots]$$

As indicated above, in this notation there will be $n-k$ columns represented with a single letter, whereas the rest of the columns in which the remaining factors can be allocated are represented as the combinations of these $n-k$ single letters. Let's consider the 16-run 2^{6-2} FF design shown in Table 16.

Table 16: 16-run Fractional Factorial design partitioned into 8 plots.

	A	B	C	D	E=ABC	F=BCD
1	-1	-1	-1	-1	-1	-1
	-1	-1	-1	1	-1	1
2	-1	-1	1	-1	1	1
	-1	-1	1	1	1	-1
3	-1	1	-1	-1	1	1
	-1	1	-1	1	1	-1
4	-1	1	1	-1	-1	-1
	-1	1	1	1	-1	1
5	1	-1	-1	-1	1	-1
	1	-1	-1	1	1	1
6	1	-1	1	-1	-1	1
	1	-1	1	1	-1	-1
7	1	1	-1	-1	-1	1
	1	1	-1	1	-1	-1
8	1	1	1	-1	1	-1
	1	1	1	1	1	1

$$[\mathbf{A}\ \ \mathbf{B}\ \ \mathbf{AB}\ \ \mathbf{C}\ \ \mathbf{AC}\ \ \mathbf{BC}\ \ \boxed{\mathbf{ABC}}\ \ \mathbf{D}\ \ \mathbf{AD}\ \ \mathbf{BD}\ \ \mathbf{ABD}\ \ \mathbf{CD}\ \ \mathbf{ACD}\ \ \boxed{\mathbf{BCD}}\ \ \mathbf{ABCD}]$$

We can also see that this design can be split into 8 WPs, with each WP having two SP runs. The 8 WPs require three *basic factors*, $[\mathrm{A} \quad \mathrm{B} \quad \mathrm{C}]$, with factor **E** being a WP design generator. The subplot design within each WP requires one *basic factor*, $[\mathrm{D}]$, with factor **F** as a SP design generator. While this design matrix helps to recognize the whole plots, when the experiment is run, the 8 WPs will be arranged in random order, and the runs within each WP will be arranged in random order. The Yates Order allows us to partition the letter groups into WP and SP factor candidates as follows:

$$\overbrace{[\underbrace{W_1\quad W_2}_{\textit{WP basic factors}}\quad W_1W_2\quad \dots}^{\text{WP Factor Candidates}}\quad \overbrace{\underbrace{S_1}_{\textit{SP basic factor}}\quad W_1S_1\quad W_2S_1\quad W_1W_2S_1\quad \dots]}^{\text{SP Factor Candidates}}$$

For these types of designs, we are only considering SPDs where the system dictates which factors should be WP factors and which should be SP factors. For further options

on choosing where to split/fractionate the design and allow SP factors to be placed at the WP level, see Bingham and Sitter (2001).

In Capehart et al. (2008), we demonstrated how to represent each letter group combination as a numeric value using a reverse form of binary conversion based upon each single letter. For example, a 32-run FF design with 5 single letters, **ABCDE**, has letter group **ABD=AB_D_**= 11010 in binary notation, which can be converted to a numeric value of 11 using reverse binary conversion. This numeric notation and conversion method matches the Yates order in ascending numeric order.

$$\begin{array}{c} [\ \mathrm{A} \quad \mathrm{B} \quad \mathrm{AB} \quad \ldots \quad \mathrm{D} \quad \mathrm{AD} \quad \mathrm{BD} \quad \mathrm{ABD} \quad \ldots\ ] \\ \downarrow \\ [\ 1 \quad 2 \quad 3 \quad \ldots \quad 8 \quad 9 \quad 10 \quad 11 \quad \ldots\ ] \end{array}$$

For FF designs, we are often interested in estimating main effects and two-factor interactions (2FIs), so it is interesting to note that this set is closed under multiplication. Thus, all 2FIs of the *k* letter groups can be simplified back into the same set. For example, if factor **D** is selected and one of the fractional generators is represented by the letter group **BCD**, their 2FI corresponds to the **(D)(BCD)=BC** letter group. Table 17 represents all 465 2FIs and the letter group they correspond to using this numbering scheme for a 32-run FF design.

Table 17: Numeric values for two-factor interactions (Row x Column) of letter groups from a 32-run FF design.

	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
1	3	2	5	4	7	6	9	8	11	10	13	12	15	14	17	16	19	18	21	20	23	22	25	24	27	26	29	28	31	30
2	-	1	6	7	4	5	10	11	8	9	14	15	12	13	18	19	16	17	22	23	20	21	26	27	24	25	30	31	28	29
3	-	-	7	6	5	4	11	10	9	8	15	14	13	12	19	18	17	16	23	22	21	20	27	26	25	24	31	30	29	28
4	-	-	-	1	2	3	12	13	14	15	8	9	10	11	20	21	22	23	16	17	18	19	28	29	30	31	24	25	26	27
5	-	-	-	-	3	2	13	12	15	14	9	8	11	10	21	20	23	22	17	16	19	18	29	28	31	30	25	24	27	26
6	-	-	-	-	-	1	14	15	12	13	10	11	8	9	22	23	20	21	18	19	16	17	30	31	28	29	26	27	24	25
7	-	-	-	-	-	-	15	14	13	12	11	10	9	8	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
8	-	-	-	-	-	-	-	1	2	3	4	5	6	7	24	25	26	27	28	29	30	31	16	17	18	19	20	21	22	23
9	-	-	-	-	-	-	-	-	3	2	5	4	7	6	25	24	27	26	29	28	31	30	17	16	19	18	21	20	23	22
10	-	-	-	-	-	-	-	-	-	1	6	7	4	5	26	27	24	25	30	31	28	29	18	19	16	17	22	23	20	21
11	-	-	-	-	-	-	-	-	-	-	7	6	5	4	27	26	25	24	31	30	29	28	19	18	17	16	23	22	21	20
12	-	-	-	-	-	-	-	-	-	-	-	1	2	3	28	29	30	31	24	25	26	27	20	21	22	23	16	17	18	19
13	-	-	-	-	-	-	-	-	-	-	-	-	3	2	29	28	31	30	25	24	27	26	21	20	23	22	17	16	19	18
14	-	-	-	-	-	-	-	-	-	-	-	-	-	1	30	31	28	29	26	27	24	25	22	23	20	21	18	19	16	17
15	-	-	-	-	-	-	-	-	-	-	-	-	-	-	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
16	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
17	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2	5	4	7	6	9	8	11	10	13	12	15	14
18	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	6	7	4	5	10	11	8	9	14	15	12	13
19	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	6	5	4	11	10	9	8	15	14	13	12
20	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3	12	13	14	15	8	9	10	11
21	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2	13	12	15	14	9	8	11	10
22	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	14	15	12	13	10	11	8	9
23	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	15	14	13	12	11	10	9	8
24	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3	4	5	6	7
25	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2	5	4	7	6
26	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	6	7	4	5
27	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	7	6	5	4
28	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1	2	3
29	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	3	2
30	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	-	1

Traditional, when an experimenter describes a blocked factorial design, they have blocking generators based on the design main effects (McLeod & Brewster, 2004; McLeod & Brewster, 2006; Montgomery, 2001). This has commonly been done for BFFSP designs as well. Instead, we propose treating the blocking factors as main effects,

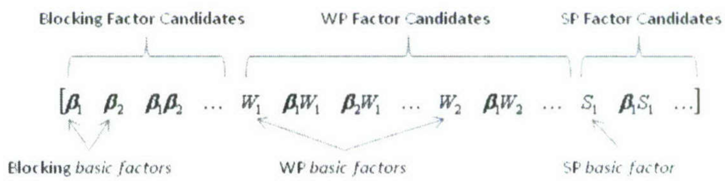
and having WP and/or SP design generators based on interactions with the blocking factors and *basic factors* from the FFSP design. McLeod and Brewster represent a BFFSP design with $2^{(n_1+n_2)-(k_1+k_2)+(b_1+b_2)}$, where n_1 and n_2 are the number of WP and SP factors, k_1 and k_2 are the number of WP and SP design generators, b_1 is the number of pure WP blocking generators, and b_2 is the number of separator blocking generators. In our research, we instead use $2^{b+(n_1+n_2)-(k_1+k_2)}$ to represent a BFFSP design where b is the number of blocking factors, n_1 and n_2 are the number of WP and SP factors, and k_1 and k_2 represent the number of WP and SP factors that are design generators. For example, McLeod and Brewster use $2^{(3+3)-(0+1)+(1+1)}$ to represent a 32-run BFFSP design with 3 WP factors, 3 SP factors, 1 pure WP block generator, 1 separation block generator, and 1 SP generator. We redefine this notation as $2^{2+(3+3)-(1+2)}$ where the first number now represents the number of blocking factors. We now have one WP generator and two SP generators.

Let's revisit the chrome-plating example to examine this difference in solution notation. McLeod and Brewster provide a design to the chrome-plating case study with $r = ABq$, $\beta_1 = ABC$, and $\delta_1 = ACpq$, for the SP main effect generator, pure WP block generator, and separator block generator respectively. This method starts with a 2^5 factorial design using [A B C p q] as WP and SP main effects and then adds three generators [$r = ABq$ $\beta_1 = ABC$ $\delta_1 = ACpq$] to form the blocked fractional factorial design. We propose representing this same design by starting with a 2^5 factorial design using [β_1 δ_1 A B p] as the blocking main effects and *basic factor* effects. Then add

three generators [$C=\beta_1 AB$ $q=\beta_1\delta_1 Bp$ $r=\beta_1\delta_1 Ap$] to form the complete blocked fractional factorial design shown below in Yates Order.

$$[\beta_1 \quad \delta_1 \quad A \quad B \quad C=\beta_1 AB \quad p \quad q=\beta_1\delta_1 Bp \quad r=\beta_1\delta_1 Ap\,]$$

Both of these designs are equivalent; however, representing the design using the later notation allows us to use the ordering scheme shown earlier in this publication and allows us to modify the IP model from Capehart et al. (2008) to find optimal BFFSP deigns. The typical running of a BFFSP design is sorted according to the following order: blocks → WPs → SPs. This fits into our partitioning scheme used on the letter groups for the FFSP designs. But now our first *basic factors* are the blocking factors.



Now that we have a representation for the set of letter groups for the blocking, WP, and SP factor candidates, let's consider the integer programming model to select the optimal design.

IP Formulation

In order to generate a BFFSP design with $2^{b+(n_1+n_2)-(k_1+k_2)}$ runs using an Integer Programming model, we define binary decision variable x_k which is equal to 1 if the k^{th} letter group is chosen as a factor. The variables x_k for $k=1,\ldots,p_1$ correspond to the blocking factors, where $p_1=2^b-1$. The variables x_k for $k=p_1+1,\ldots,p_1+p_2$ correspond to the WP factors, with $p_2=2^{b+(n_1-k_1)}-1$. And the variables x_k for

$k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3$ correspond to the SP factors with $p_3 = 2^{b+(n_1+n_2)-(k_1+k_2)} - 1$. Binary variable w_{ij} represents 2FIs between letter groups i and j, i.e. $w_{ij} = 1$ if and only if $x_i = 1$ and $x_j = 1$. We also introduce the binary variable y^k which is equal to 1 if exactly one 2FI equivalent to the k^{th} letter group is chosen. If there is more than one 2FI that corresponds to the k^{th} letter group then the integer variable z^k will give this number. The following constraints provide an appropriate BFFSP IP model:

$$\sum_{k=1}^{p_1} x_k = b \tag{3.1.1}$$

$$\sum_{k=p_1+1}^{p_1+p_2} x_k = n_1 \tag{3.1.2}$$

$$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} x_k = n_2 \tag{3.1.3}$$

Constraints (3.1.1), (3.1.2), and (3.1.3) set the number of blocking and experiment main effects.

$$x_i + x_j - w_{ij} \leq 1,\ p_1 < i < j \tag{3.2.1}$$

$$x_i + x_j - 2w_{ij} \geq 0,\ p_1 < i < j \tag{3.2.2}$$

Constraints (3.2.1) and (3.2.2) enforce that $w_{ij} = 1$ if and only if $x_i = 1$ and $x_j = 1$, representing a 2FI.

$$\sum_{ij \in S^k} w_{ij} = y_k + z_k\ ,\ k = p_1 + 1, \ldots, p_1 + p_2 + p_3 \tag{3.3.1}$$

$$z_k \le M(1-y_k),\, k = p_1+1,\ldots,p_1+p_2+p_3 \qquad (3.3.2)$$

$$y_k + z_k \le M\tau_k\,,\, k = p_1+1,\ldots,p_1+p_2+p_3 \qquad (3.3.3)$$

Constraints (3.3.1) and (3.3.2) represent the 2FIs corresponding to each letter group. $\boldsymbol{S}^k$ is the set of 2FIs that are equivalent to the k^{th} letter group. Here, y^k will be equal to 1 if we select exactly one 2FI equivalent to the k^{th} letter group and z^k will give the number of 2FIs if there is more than one 2FI selected that corresponds to the k^{th} letter group. Constraint (3.3.3) sets binary variable $\tau^k = 1$ if there is at least one 2FI equivalent to the k^{th} letter group.

$$(1-\tau_k)+x_k-\tau_k^{\text{WP}} \le 1\,,\, k = p_1+1,\ldots,p_1+p_2 \qquad (3.4.1)$$

$$(1-\tau_k)+x_k-2\tau_k^{\text{WP}} \ge 0\,,\, k = p_1+1,\ldots,p_1+p_2 \qquad (3.4.2)$$

$$\sum_{k=p_1+1}^{p_1+p_2}\tau_k^{\text{WP}} = ClearWPs \qquad (3.4.3)$$

Constraints (3.4.1) and (3.4.2) are used to determine which WP factors are clear. We set $\tau_k^{\text{WP}} = 1$ if and only if we select the k^{th} letter group, $x_k = 1$, and there are no 2FIs equivalent to that same letter group, $(1-\tau_k)=1$. Constraint (3.4.3) then sums up the number of clear WP effects.

$$(1-\tau_k)+x_k-\tau_k^{\text{SP}} \le 1\,,\, k = p_1+p_2+1,\ldots,p_1+p_2+p_3 \qquad (3.5.1)$$

$$(1-\tau_k)+x_k-2\tau_k^{\text{SP}} \ge 0\,,\, k = p_1+p_2+1,\ldots,p_1+p_2+p_3 \qquad (3.5.2)$$

$$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} \tau_k^{\mathrm{SP}} = ClearSPs \qquad (3.5.3)$$

Similarly, constraints (3.5.1) and (3.5.2) look at the SP factors, setting binary variable $\tau_k^{\mathrm{SP}} = 1$ if and only if we select the k^{th} letter group, $x_k = 1$, and there are no 2FIs equivalent to that same letter group, $(1-\tau_k) = 1$. Constraint (3.5.3) calculates the total number of clear SP factors.

$$\sum_{ij \in S_{WW}^k} w_{ij} = y_k^{\mathrm{WW}} + z_k^{\mathrm{WW}},\ k = p_1+1,\ldots,p_1+p_2 \qquad (3.6.1)$$

$$z_k^{\mathrm{WW}} \le M\left(1-y_k^{\mathrm{WW}}\right),\ k = p_1+1,\ldots,p_1+p_2 \qquad (3.6.2)$$

$$y_k^{\mathrm{WW}} + y_k + (1-x_k) - \tau_k^{\mathrm{WW}} \le 2,\ k = p_1+1,\ldots,p_1+p_2 \qquad (3.6.3)$$

$$y_k^{\mathrm{WW}} + y_k + (1-x_k) - 3\tau_k^{\mathrm{WW}} \ge 0,\ k = p_1+1,\ldots,p_1+p_2 \qquad (3.6.4)$$

$$\sum_{k=p_1+1}^{p_1+p_2} \tau_k^{\mathrm{WW}} = WPWP2FIs \qquad (3.6.5)$$

Constraints (3.6.1) and (3.6.2) work in a similar manner to those of (3.3.1) and (3.3.2), with S_{WW}^k representing the set of 2FIs involving two WP factors that are equivalent to the k^{th} letter group. Here, we are using the binary variable $y_k^{\mathrm{WW}} = 1$ when there is a 2FI consisting of two WP effects that is equivalent to the k^{th} letter group. Constraints (3.6.3) and (3.6.4) set binary variable $\tau_k^{\mathrm{WW}} = 1$ when there is only one overall 2FI equivalent to the k^{th} letter group (y_k) and that 2FI consists of two WP factors (y_k^{WW}), and also when there is no main effect selected for the k^{th} letter group (x_k). Finally, constraint (3.6.5)

sums across all k letter groups for WP factors to determine the number of clear WPxWP 2FIs.

$$\sum_{ij \in S_{SS}^k} w_{ij} = y_k^{ss} + z_k^{ss}, k = p_1 + 1, \ldots, p_1 + p_2 + p_3 \tag{3.7.1}$$

$$z_k^{ss} \leq M\left(1 - y_k^{ss}\right), k = p_1 + 1, \ldots, p_1 + p_2 + p_3 \tag{3.7.2}$$

$$y_k^{ss} + y_k + \left(1 - x_k\right) - \tau_k^{ss} \leq 2, k = p_1 + 1, \ldots, p_1 + p_2 + p_3 \tag{3.7.3}$$

$$y_k^{ss} + y_k + \left(1 - x_k\right) - 3\tau_k^{ss} \geq 0, k = p_1 + 1, \ldots, p_1 + p_2 + p_3 \tag{3.7.4}$$

$$\sum_{k=p_1+1}^{p_1+p_2+p_3} \tau_k^{ss} = SPSP2FIs \tag{3.7.5}$$

$$\sum_{k=p_1+1}^{p_1+p_2} \tau_k^{ss} = SPSP2FIs\_WPerror \tag{3.7.6}$$

Type (3.7) constraints work in the same manner as those of type (3.6) with S_{SS}^k representing the set of 2FIs involving two SP factors that are equivalent to the k^{th} letter group. An additional constraint (3.7.6) is added to count the number of clear SP 2FIs that are tested against the WP error. As pointed out by Bisgaard (2000), SP interactions that are aliased with WP main effects or interactions involving only WP factors are tested against the WP error. From our numbering scheme, we know that WP main effects and interactions involving only WP factors refers to letter groups $k = p_1 + 1, \ldots, p_1 + p_2$.

$$\sum_{ij \in S_{WS}^k} w_{ij} = y_k^{ws} + z_k^{ws}, k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3 \tag{3.8.1}$$

$$z_k^{ws} \leq M\left(1 - y_k^{ws}\right), k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3 \tag{3.8.2}$$

$$y_k^{ws} + y_k + (1 - x_k) - \tau_k^{ws} \le 2,\ k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3 \quad (3.8.3)$$

$$y_k^{ws} + y_k + (1 - x_k) - 3\tau_k^{ws} \ge 0,\ k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3 \quad (3.8.4)$$

$$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} \tau_k^{ws} = WPSP2FIs \quad (3.8.5)$$

We determine the number of clear 2FIs involving one WP factor and one SP factor using similar constraints to type (3.6) and (3.7), except with S_{WS}^k representing the set of 2FIs involving one WP factor and one SP factor and equivalent to the k^{th} letter group.

Finally, we add constraint (3.9) to set the *basic factors*, including the blocking factors.

$$x_{2^{m-1}} = 1, m = 1, \ldots, b + (n_1 + n_2) - (k_1 + k_2) \quad (3.9)$$

We now can customize the objective function based on various criteria using the variables: ***clearWPs***, ***clearSPs***, ***WPWP2FIs***, ***SPSP2FIs***, and ***WPSP2FIs***. For example, if we want to maximize the number of clear 2FIs, but we're not concerned about whether the design is Resolution III or Resolution IV, we would have the following objective function:

$$\text{Max } Z = (WPWP2FIs + SPSP2FIs + WPSP2FIs)$$

Or we could maximize the number of clear SP factors and the 2FIs involving SP factors by using this objective function:

$$\text{Max } Z = clearSPs + (SPSP2FIs + WPSP2FIs)$$

This last example has equal importance for one clear SP main effect or any 2FI involving a SP factor. If we want to make the number of clear SP factors a higher priority, we would use weights in the objective function as follows:

$$\text{Max } Z = 10 \times clearSPs + (SPSP2FIs + WPSP2FIs)$$

McLeod and Brewster (2004) provide tables with secondary criteria after first looking for a MA design. The first five secondary criteria are as follows, with decreasing importance:

1. Number of clear main effects
2. Number of clear 2FIS
3. Number of clear SP main effects
4. Number of clear SP 2FIs
5. Number of clear SP 2FIs tested against WP error

Here is how can represent these criteria using a form of goal programming:

$$\text{Max} \begin{array}{l} Z = 10000 \times (clearWPs + clearSPs) + 1000 \times (WPWP2FIs + SPSP2FIs + WPSP2FIs) \\ \quad + 100 \times clearSPs + 10 \times (SPSP2FIs + WPSP2FIs) + SPSP2FIs\_WPerror \end{array}$$

Example 1: 16-run, 2 blocks, 4 WPs per block, 2 SP runs per WP

Case 1: 3 WP factors and 2 SP factors

Here we have a BFFSP design with *structure* 2:4:2. First assume that we are selecting 3 WP factors and 2 SP factors, so that we are looking for a $2^{1+(3+2)-(1+1)}$ design. Let's look at how each type of constraint is represented for this specific problem.

Table 18: List of equations for BFFSP design with 2:4:2 structure, 3 WP factors and 2 SP factors.

Constraint Type	Formula	Number of constraints
(3.1.1)	$\sum_{k=1}^{1} x_k = 1$	1
(3.1.2)	$\sum_{k=2}^{7} x_k = 3$	1

Constraint Type	Formula	Number of constraints
(3.1.3)	$\sum_{k=8}^{15} x_k = 2$	1
(3.2.1)	$x_i + x_j - w_{ij} \le 1, 1 < i < j$	91
(3.2.2)	$x_i + x_j - 2w_{ij} \ge 0, 1 < i < j$	91
(3.3.1)	$\sum_{ij \in S^k} w_{ij} = y_k + z_k, k = 2,\ldots,15$	14
(3.3.2)	$z_k \le M(1 - y_k), k = 2,\ldots,15$	14
(3.3.3)	$y_k + z_k \le M\tau_k, k = 2,\ldots,15$	14
(3.4.1)	$(1-\tau_k) + x_k - \tau_k^{WP} \le 1, k = 2,\ldots,7$	6
(3.4.2)	$(1-\tau_k) + x_k - 2\tau_k^{WP} \ge 0, k = 2,\ldots,7$	6
(3.4.3)	$\sum_{k=2}^{7} \tau_k^{WP} = ClearWPs$	1
(3.5.1)	$(1-\tau_k) + x_k - \tau_k^{SP} \le 1, k = 8,\ldots,15$	8
(3.5.2)	$(1-\tau_k) + x_k - 2\tau_k^{SP} \ge 0, k = 8,\ldots,15$	8
(3.5.3)	$\sum_{k=8}^{15} \tau_k^{SP} = ClearSPs$	1
(3.6.1)	$\sum_{ij \in S_{WW}^k} w_{ij} = y_k^{WW} + z_k^{WW}, k = 2,\ldots,7$	6
(3.6.2)	$z_k^{WW} \le M(1 - y_k^{WW}), k = 2,\ldots,7$	6
(3.6.3)	$y_k^{WW} + y_k + (1 - x_k) - \tau_k^{WW} \le 2, k = 2,\ldots,7$	6
(3.6.4)	$y_k^{WW} + y_k + (1 - x_k) - 3\tau_k^{WW} \ge 0, k = 2,\ldots,7$	6
(3.6.5)	$\sum_{k=2}^{7} \tau_k^{WW} = WPWP2FIs$	1
(3.7.1)	$\sum_{ij \in S_{SS}^k} w_{ij} = y_k^{SS} + z_k^{SS}, k = 2,\ldots,15$	14
(3.7.2)	$z_k^{SS} \le M(1 - y_k^{SS}), k = 2,\ldots,15$	14
(3.7.3)	$y_k^{SS} + y_k + (1 - x_k) - \tau_k^{SS} \le 2, k = 2,\ldots,15$	14
(3.7.4)	$y_k^{SS} + y_k + (1 - x_k) - 3\tau_k^{SS} \ge 0, k = 2,\ldots,15$	14
(3.7.5)	$\sum_{k=2}^{15} \tau_k^{SS} = SPSP2FIs$	1
(3.7.6)	$\sum_{k=2}^{7} \tau_k^{SS} = SPSP2FIs\_WPerror$	1
(3.8.1)	$\sum_{ij \in S_{WS}^k} w_{ij} = y_k^{WS} + z_k^{WS}, k = 8,\ldots,15$	8

Constraint Type	Formula	Number of constraints
(3.8.2)	$z_k^{ws} \le M(1-y_k^{ws}), k=8,\ldots,15$	8
(3.8.3)	$y_k^{ws}+y_k+(1-x_k)-\tau_k^{ws} \le 2, k=8,\ldots,15$	8
(3.8.4)	$y_k^{ws}+y_k+(1-x_k)-3\tau_k^{ws} \ge 0, k=8,\ldots,15$	8
(3.8.5)	$\sum_{k=8}^{15}\tau_k^{ws} = WPSP2FIs$	1
(3.9)	$x_{2^{m-1}} = 1, m=1,\ldots,4$	4

Constraints (3.3.1) involve the binary variable w_{ij}. This refers to which k^{th} letter group each 2FI maps back into. We use Table 17 to build the mapping in Table 19 for each 2FI, disregarding mapping into the block letter group ($k=1$) or any 2FIs involving the blocking factor (i.e. $w_{1,2}$, $w_{1,3}$, etc.).

Table 19: 2FI mapping for main effects. Does not include Blocking effects or Blocking x Blocking main effects.

k		Main	2FI					
2	WP Factors	x_2	$w_{4,6}$	$w_{5,7}$	$w_{8,10}$	$w_{9,11}$	$w_{12,14}$	$w_{13,15}$
3		x_3	$w_{4,7}$	$w_{5,6}$	$w_{8,11}$	$w_{9,10}$	$w_{12,15}$	$w_{13,14}$
4		x_4	$w_{2,6}$	$w_{3,7}$	$w_{8,12}$	$w_{9,13}$	$w_{10,14}$	$w_{11,15}$
5		x_5	$w_{2,7}$	$w_{3,6}$	$w_{8,13}$	$w_{9,12}$	$w_{10,15}$	$w_{11,14}$
6		x_6	$w_{2,4}$	$w_{3,5}$	$w_{8,14}$	$w_{9,15}$	$w_{10,12}$	$w_{11,13}$
7		x_7	$w_{2,5}$	$w_{3,4}$	$w_{8,15}$	$w_{9,14}$	$w_{10,13}$	$w_{11,12}$
8	SP Factors	x_8	$w_{2,10}$	$w_{3,11}$	$w_{4,12}$	$w_{5,13}$	$w_{6,14}$	$w_{7,15}$
9		x_9	$w_{2,11}$	$w_{3,10}$	$w_{4,13}$	$w_{5,12}$	$w_{6,15}$	$w_{7,14}$
10		x_{10}	$w_{2,8}$	$w_{3,9}$	$w_{4,14}$	$w_{5,15}$	$w_{6,12}$	$w_{7,13}$
11		x_{11}	$w_{2,9}$	$w_{3,8}$	$w_{4,15}$	$w_{5,14}$	$w_{6,13}$	$w_{7,12}$
12		x_{12}	$w_{2,14}$	$w_{3,15}$	$w_{4,8}$	$w_{5,9}$	$w_{6,10}$	$w_{7,11}$
13		x_{13}	$w_{2,15}$	$w_{3,14}$	$w_{4,9}$	$w_{5,8}$	$w_{6,11}$	$w_{7,10}$
14		x_{14}	$w_{2,12}$	$w_{3,13}$	$w_{4,10}$	$w_{5,11}$	$w_{6,8}$	$w_{7,9}$
15		x_{15}	$w_{2,13}$	$w_{3,12}$	$w_{4,11}$	$w_{5,10}$	$w_{6,9}$	$w_{7,8}$

Table 19 also demonstrates a property that distinguishes the mapping of WPxWP, SPxSP, and WPxWP 2FIs. Note that WPxWP 2FIs (top-left 2FI section) map back into WP letter groups, while WPxSP 2FIs (bottom 2FI section) map into SP letter groups. SPxSP 2FIs can map into either WP or SP letter groups. This mapping property for 2FIs allows us adjust the summations for several of the IP constraints.

Now let's use this example above with a specific goal. We will try to maximize the number of clear 2FIs by using the following objective function:

$$\text{Max } Z = (WPWP2FIs + SPSP2FIs + WPSP2FIs)$$

Our IP formulation provides the following results [β_1, A, B, $C = \beta_1 AB$, p, $q = \beta_1 p$]. These correspond to letter groups [1, 2, 4, 7, 8, 9]. The corresponding 2FIs, not including the blocking factor are [(2,4), (2,7), (2,8), (2,9), (4,7), (4,8), (4,9), (7,8), (7,9), (8,9)]

Table 20: 2FI mapping for BFFSPD with 2:4:4 structure and 3/2 WP/SP factors. Maximizing number of clear 2FIs.

k		Main	2FI					
2	WP Factors	x_2	$w_{4,6}$	$w_{5,7}$	$w_{8,10}$	$w_{9,11}$	$w_{12,14}$	$w_{13,15}$
3		x_3	$w_{4,7}$	$w_{5,6}$	$w_{8,11}$	$w_{9,10}$	$w_{12,15}$	$w_{13,14}$
4		x_4	$w_{2,6}$	$w_{3,7}$	$w_{8,12}$	$w_{9,13}$	$w_{10,14}$	$w_{11,15}$
5		x_5	$w_{2,7}$	$w_{3,6}$	$w_{8,13}$	$w_{9,12}$	$w_{10,15}$	$w_{11,14}$
6		x_6	$w_{2,4}$	$w_{3,5}$	$w_{8,14}$	$w_{9,15}$	$w_{10,12}$	$w_{11,13}$
7		x_7	$w_{2,5}$	$w_{3,4}$	$w_{8,15}$	$w_{9,14}$	$w_{10,13}$	$w_{11,12}$
8	SP Factors	x_8	$w_{2,10}$	$w_{3,11}$	$w_{4,12}$	$w_{5,13}$	$w_{6,14}$	$w_{7,15}$
9		x_9	$w_{2,11}$	$w_{3,10}$	$w_{4,13}$	$w_{5,12}$	$w_{6,15}$	$w_{7,14}$
10		x_{10}	$w_{2,8}$	$w_{3,9}$	$w_{4,14}$	$w_{5,15}$	$w_{6,12}$	$w_{7,13}$
11		x_{11}	$w_{2,9}$	$w_{3,8}$	$w_{4,15}$	$w_{5,14}$	$w_{6,13}$	$w_{7,12}$
12		x_{12}	$w_{2,14}$	$w_{3,15}$	$w_{4,8}$	$w_{5,9}$	$w_{6,10}$	$w_{7,11}$
13		x_{13}	$w_{2,15}$	$w_{3,14}$	$w_{4,9}$	$w_{5,8}$	$w_{6,11}$	$w_{7,10}$
14		x_{14}	$w_{2,12}$	$w_{3,13}$	$w_{4,10}$	$w_{5,11}$	$w_{6,8}$	$w_{7,9}$
15		x_{15}	$w_{2,13}$	$w_{3,12}$	$w_{4,11}$	$w_{5,10}$	$w_{6,9}$	$w_{7,8}$

We can see from the table above that there are 9 clear 2FIs. Three of them are WPxWP 2FIs and 6 are WPxSP 2FIs. Plus, it is easy to tell that the 2FIs are clear from the 5 main effects.

Case 2: 4 WP factors and 2 SP factors

Now if we decide to have 4 WP factors and 2 SP factors, still with 2 blocks in the design, and still use the same objective function, our factors become [β_1, A, B, $C = AB$, $D = \beta_1 AB$, p, $q = \beta_1 ABp$]. These correspond to letter groups [1, 2, 4, 6, 7, 8, 15]. The corresponding 2FIs, not including the blocking factor are [(2,4), (2,6), (2,7), (2,8), (2,15), (4,6), (4,7), (4,8), (4,15), (6,7), (6,8), (6,15), (7,8), (7,15), (8,15)]

Table 21: 2FI mapping for BFFSPD with 2:4:4 structure and 4/2 WP/SP factors. Still maximizing number of clear 2FIs.

k		Main	2FI					
2	WP Factors	x_2	$w_{4,6}$	$w_{5,7}$	$w_{8,10}$	$w_{9,11}$	$w_{12,14}$	$w_{13,15}$
3		x_3	$w_{4,7}$	$w_{5,6}$	$w_{8,11}$	$w_{9,10}$	$w_{12,15}$	$w_{13,14}$
4		x_4	$w_{2,6}$	$w_{3,7}$	$w_{8,12}$	$w_{9,13}$	$w_{10,14}$	$w_{11,15}$
5		x_5	$w_{2,7}$	$w_{3,6}$	$w_{8,13}$	$w_{9,12}$	$w_{10,15}$	$w_{11,14}$
6		x_6	$w_{2,4}$	$w_{3,5}$	$w_{8,14}$	$w_{9,15}$	$w_{10,12}$	$w_{11,13}$
7		x_7	$w_{2,5}$	$w_{3,4}$	$w_{8,15}$	$w_{9,14}$	$w_{10,13}$	$w_{11,12}$
8	SP Factors	x_8	$w_{2,10}$	$w_{3,11}$	$w_{4,12}$	$w_{5,13}$	$w_{6,14}$	$w_{7,15}$
9		x_9	$w_{2,11}$	$w_{3,10}$	$w_{4,13}$	$w_{5,12}$	$w_{6,15}$	$w_{7,14}$
10		x_{10}	$w_{2,8}$	$w_{3,9}$	$w_{4,14}$	$w_{5,15}$	$w_{6,12}$	$w_{7,13}$
11		x_{11}	$w_{2,9}$	$w_{3,8}$	$w_{4,15}$	$w_{5,14}$	$w_{6,13}$	$w_{7,12}$
12		x_{12}	$w_{2,14}$	$w_{3,15}$	$w_{4,8}$	$w_{5,9}$	$w_{6,10}$	$w_{7,11}$
13		x_{13}	$w_{2,15}$	$w_{3,14}$	$w_{4,9}$	$w_{5,8}$	$w_{6,11}$	$w_{7,10}$
14		x_{14}	$w_{2,12}$	$w_{3,13}$	$w_{4,10}$	$w_{5,11}$	$w_{6,8}$	$w_{7,9}$
15		x_{15}	$w_{2,13}$	$w_{3,12}$	$w_{4,11}$	$w_{5,10}$	$w_{6,9}$	$w_{7,8}$

There are 8 clear 2FIs. Note there are no clear main effects.

If we change our objective function so that we are first interested in clear main effects, and then clear 2FIs. The objective function becomes:

$$\text{Max } Z = 10\times(clearWPs + clearSPs) + (WPWP2FIs + SPSP2FIs + WPSP2FIs)$$

Our factors become [β_1, A, B, $C = \beta_1 A$, $D = \beta_1 B$, p, $q = ABp$]. These correspond to letter groups [1, 2, 3, 4, 5, 8, 14]. The corresponding 2FIs, not including the blocking factor are [(2,3), (2,4), (2,5), (2,8), (2,14), (3,4), (3,5), (3,8), (3,14), (4,5), (4,8), (4,14), (5,8), (5,14), (8,14)]

Table 22: 2FI mapping for BFFSPD with 2:4:4 structure and 4/2 WP/SP factors. Prioritized objective to maximize number of clear main effects and then 2FIs.

k		Main	2FI					
2	WP Factors	x_2	$w_{4,6}$	$w_{5,7}$	$w_{8,10}$	$w_{9,11}$	$w_{12,14}$	$w_{13,15}$
3		x_3	$w_{4,7}$	$w_{5,6}$	$w_{8,11}$	$w_{9,10}$	$w_{12,15}$	$w_{13,14}$
4		x_4	$w_{2,6}$	$w_{3,7}$	$w_{8,12}$	$w_{9,13}$	$w_{10,14}$	$w_{11,15}$
5		x_5	$w_{2,7}$	$w_{3,6}$	$w_{8,13}$	$w_{9,12}$	$w_{10,15}$	$w_{11,14}$
6		x_6	$w_{2,4}$	$w_{3,5}$	$w_{8,14}$	$w_{9,15}$	$w_{10,12}$	$w_{11,13}$
7		x_7	$w_{2,5}$	$w_{3,4}$	$w_{8,15}$	$w_{9,14}$	$w_{10,13}$	$w_{11,12}$
8	SP Factors	x_8	$w_{2,10}$	$w_{3,11}$	$w_{4,12}$	$w_{5,13}$	$w_{6,14}$	$w_{7,15}$
9		x_9	$w_{2,11}$	$w_{3,10}$	$w_{4,13}$	$w_{5,12}$	$w_{6,15}$	$w_{7,14}$
10		x_{10}	$w_{2,8}$	$w_{3,9}$	$w_{4,14}$	$w_{5,15}$	$w_{6,12}$	$w_{7,13}$
11		x_{11}	$w_{2,9}$	$w_{3,8}$	$w_{4,15}$	$w_{5,14}$	$w_{6,13}$	$w_{7,12}$
12		x_{12}	$w_{2,14}$	$w_{3,15}$	$w_{4,8}$	$w_{5,9}$	$w_{6,10}$	$w_{7,11}$
13		x_{13}	$w_{2,15}$	$w_{3,14}$	$w_{4,9}$	$w_{5,8}$	$w_{6,11}$	$w_{7,10}$
14		x_{14}	$w_{2,12}$	$w_{3,13}$	$w_{4,10}$	$w_{5,11}$	$w_{6,8}$	$w_{7,9}$
15		x_{15}	$w_{2,13}$	$w_{3,12}$	$w_{4,11}$	$w_{5,10}$	$w_{6,9}$	$w_{7,8}$

We can see from the table above that there are no clear 2FIs. However, all 6 main effects are clear. This was due to the prioritization and weighting scheme in the objective function.

Let's now look at another variation to the objective function. Considering the same number of factors, let's change our objective function so that we first would like to have clear SP factors and then 2FIs involving SP factors. The objective function becomes:

$$\text{Max } Z = 10 \times clearSPs + (SPSP2FIs + WPSP2FIs)$$

Our factors will be [β_1, A, B, $C = \beta_1 A$, $D = AB$, p, $q = \beta_1 ABp$]. These correspond to letter groups [1, 2, 3, 4, 6, 8, 15]. The corresponding 2FIs, not including the blocking

factor are [(2,3), (2,4), (2,6), (2,8), (2,15), (3,4), (3,6), (3,8), (3,15), (4,6), (4,8), (4,15), (6,8), (6,15), (8,15)]

Table 23: 2FI mapping for BFFSPD with 2:4:4 structure and 4/2 WP/SP factors. First maximize number of clear SP main effects and then 2FIs involving SP factors.

k		Main	2FI					
2	WP Factors	x_2	$w_{4,6}$	$w_{5,7}$	$w_{8,10}$	$w_{9,11}$	$w_{12,14}$	$w_{13,15}$
3		x_3	$w_{4,7}$	$w_{5,6}$	$w_{8,11}$	$w_{9,10}$	$w_{12,15}$	$w_{13,14}$
4		x_4	$w_{2,6}$	$w_{3,7}$	$w_{8,12}$	$w_{9,13}$	$w_{10,14}$	$w_{11,15}$
5		x_5	$w_{2,7}$	$w_{3,6}$	$w_{8,13}$	$w_{9,12}$	$w_{10,15}$	$w_{11,14}$
6		x_6	$w_{2,4}$	$w_{3,5}$	$w_{8,14}$	$w_{9,15}$	$w_{10,12}$	$w_{11,13}$
7		x_7	$w_{2,5}$	$w_{3,4}$	$w_{8,15}$	$w_{9,14}$	$w_{10,13}$	$w_{11,12}$
8	SP Factors	x_8	$w_{2,10}$	$w_{3,11}$	$w_{4,12}$	$w_{5,13}$	$w_{6,14}$	$w_{7,15}$
9		x_9	$w_{2,11}$	$w_{3,10}$	$w_{4,13}$	$w_{5,12}$	$w_{6,15}$	$w_{7,14}$
10		x_{10}	$w_{2,8}$	$w_{3,9}$	$w_{4,14}$	$w_{5,15}$	$w_{6,12}$	$w_{7,13}$
11		x_{11}	$w_{2,9}$	$w_{3,8}$	$w_{4,15}$	$w_{5,14}$	$w_{6,13}$	$w_{7,12}$
12		x_{12}	$w_{2,14}$	$w_{3,15}$	$w_{4,8}$	$w_{5,9}$	$w_{6,10}$	$w_{7,11}$
13		x_{13}	$w_{2,15}$	$w_{3,14}$	$w_{4,9}$	$w_{5,8}$	$w_{6,11}$	$w_{7,10}$
14		x_{14}	$w_{2,12}$	$w_{3,13}$	$w_{4,10}$	$w_{5,11}$	$w_{6,8}$	$w_{7,9}$
15		x_{15}	$w_{2,13}$	$w_{3,12}$	$w_{4,11}$	$w_{5,10}$	$w_{6,9}$	$w_{7,8}$

The two SP factors are clear, and there are four clear WPxSP 2FIs. Although there is also a WP factor and WPxWP 2FI clear, we were not concerned with these since they did not add to the objective function.

Example 2: Case study revisited

Now that we have an understanding of how to model the BFFSP design using IP formulation, let's return to the chrome-plating example from McLeod and Brewster (2004). Once again, this case has a 4:4:2 structure. Thus, we are looking for 2 basic

blocking factors, 2 basic WP factors, and 1 basic SP factor. The remaining WP factor and two subplot factors in the design are design generators. Table 24 below compares their optimal BFFSP designs for screening (MA) and robust parameter design (RPD) in the chrome-plating experiment with our IP formulation design (IP) using the top four secondary criteria from (2004) and top three secondary criteria from (2006). The remaining secondary criteria in their publications dealt with number of effects tested against error terms. That type of analysis is beyond the scope of this paper.

Table 24: Three different SPDs for the chrome-plating case study, and how they score using the secondary criteria from McLeod and Brewster.

				McLeod/Brewster(2004)				McLeod/Brewster(2006)		
n1, n2	*Structure*	*Design*	*Design Generators*	*(a)*	*(b)*	*(c)*	*(d)*	*(a)*	*(b)*	*(c)*
MA: 3,3	4:4:2	3,3;0,1;1,1	$ABC\beta_1$, Abqr, $Acpq\delta_1$	6	9	3	7	3	5	2
RPD: 3,3	4:4:2	3,3;1,0;0,2	ABC, $Apq\delta_1$, $Bpr\delta_2$	3	12	3	12	3	9	3
IP: 3,3	4:4:2	2;3,3;1,2	$AC\beta_1$, $Bpq\beta_1$, $Abpr\beta_1\beta_2$	6	14	3	12	3	9	3

We can see that our IP optimal BFFSP design meets or exceeds both the MA and RPD optimal designs. McLeod and Brewster describe a $2^{(n_1+n_2)-(k_1+k_2)+(b_1+b_2)}$ design as "Design = n_1,n_2,k_1,k_2,b_1,b_2" where n_1 and n_2 are the number of WP and SP factors, k_1 and k_2 are the number of WP and SP design generators, b_1 is the number of pure blocking generators, and b_2 is the number of "separator" blocking generators. In this research, we instead use "Design = b,n_1,n_2,k_1,k_2" where b is the number of blocking factors, n_1 and n_2 are the number of WP and SP factors, and k_1 and k_2 are the number of WP and SP design generators. A pure blocking generator is available by reducing the WP design by a fraction, and a "separation" blocking generator is available by reducing the SP design by a fraction. However, our formulation allows for a type of design not considered by

McLeod and Brewster; where the blocking generator is both a pure and a "separation" generator. We will show this in the next example.

Example 3: Comparison with McLeod and Brewster (2004)

Let's look at five other cases from literature to see how this IP formulation compares.

Table 25: Comparisons of various BFFSP designs generated using the IP model to those from McLeod/Brewster (2004).

Case	*n1, n2*	*Structure*	*Design (Brewster/McLeod)*	*Design (Capehart)*	*(a)*	*(b)*	*(c)*	*(d)*	*(e)*	*WLP*
1	3,4	4:2:4	3,4,0,2,2,0		7	12	4	12	1	0,3,1,4,2,2,0,2,0,1
				2,3,4,2,2	7	13	4	13	0	0,4,1,3,2,2,0,2,0,0,1
2	3,4	4:4:2	3,4,0,2,1,1		7	6	4	4	0	0,0,3,7,0,4,0,0,0,0,1
				2,3,4,1,3	7	15	4	13	5	0,2,1,5,2,4,0,0,0,0,1
3	3,4	4:4:2	3,4,1,1,0,2		4	12	4	12	3	1,0,1,6,0,6,0,0,1
				2,3,4,1,3	7	15	4	13	5	0,2,1,5,2,4,0,0,0,0,1
4	4,4	4:4:2	4,4,0,3,2,0		8	12	4	8	0	0,1,3,10,4,8,0,0,0,3,0,2
				2,4,4,2,3	8	13	4	10	3	0,6,3,3,4,6,0,6,0,0,0,3
5	7,2	2:8:2	7,2,3,1,1,0		9	2	2	2	0	0,0,10,8,0,0,4,4,0,0,1,4
				1,7,2,4,1	9	15	2	15	1	0,3,7,1,7,4,0,4,0,1,0,3,1

What we have here are five examples from McLeod and Brewster (2004), where criteria (a) – (e) are as follows:

a) Number of clear main effects

b) Number of clear 2FIs

c) Number of clear SP main effects

d) Number of clear SP 2FIs

e) Number of clear SP 2FIs tested against WP error

McLeod and Brewster used these criteria in decreasing order of importance after first requiring the design to be MA. We do not consider the MA criterion, but instead use goal programming in the objective function to accommodate the four criteria. Larger

numbers for goals (a) – (d) are preferred. The IP designs beat those of McLeod and Brewster in all five cases.

Looking at case 2 and 3 in Table 25, we see that our proposed IP model results in the same design for both cases:

$$[\beta_1 \;\; \beta_2 \;\; \mathrm{A} \;\; \mathrm{B} \;\; \mathrm{C}=\beta_2\mathrm{A} \;\; \mathrm{p} \;\; q=\beta_2 p \;\; r=\beta_1\mathrm{A}p \;\; s=\beta_1\beta_2\mathrm{AB}p\,]$$

In this design, it is not difficult to see that $\beta_2 = \mathrm{AC} = pq$, which is both a pure and "separation" blocking generator. This type of blocking generator is different from the ones discussed by McLeod and Brewster.

Conclusion

We have discussed a new approach to representing and designing optimal BFFSP designs by using the binary representation of their letter notation to incorporate integer programming techniques. This eliminates the need to reference the design matrix in order to make calculations when analyzing the alias structure for competing designs. By breaking up the components of clear main effects and clear 2FIs, we allow the experimenter to customize their objective function to best suit their needs.

CHAPTER 4: GENERATING SPLIT-PLOT DESIGNS FOR MULTIPLE STEPS

We have examined manufacturing processes that involve factors that are hard-to-change and factors that are easy-to-change. Often, it is not the difficulty to change the factor that puts it into a separate category, but rather the location along the production line in which that factor has an effect on the final product. Many complex manufacturing production lines involve a product going through several sequential processing steps. The output characteristics of this product do not merely reflect the effect of the factors of the current step in the process, but also on preceding steps in the process. Most designed experiments are focused on a single process, making analyzing multiple process steps during one experiment both unusual and more complex. The split-plot design structure can be used for examining systems involving multiple processing steps.

The objective of this chapter is to generate a first-order experimental design using a split-plot structure to examine the relationship between factors for multiple sequential processes. A process with m steps can be represented by an m-stage split-plot design. And in turn, since each step in the process may have multiple factors that affect the product, each level of the split-plot design will have multiple factors to manipulate. The first step in the process, and its associated factors, comprise the whole plot. Each preceding step corresponds to a subplot in the split plot design. For this research, we will incorporate the work from our previous two chapters to create an integer programming model that can create FFSP designs with more than two stages.

Increasing the IP Model for 3-Stages

A simple approach is to work with the IP model from Chapter 3 and treat the blocking factors as the first stage factors. Although, since we are no longer ignoring

interactions between blocking factors and regular factors, we must adjust some of the constraints. Table 26 lists the set of constraints for a 3-Staged FFSP design having $2^{(n_1+n_2+n_3)-(k_1+k_2+k_3)}$ runs. There are n_i and k_i factors and design generators repectively for each i^{th} stage. The variables x_k for $k=1,\ldots,p_1$ correspond to the stage-1 candidate factors, where $p_1 = 2^{n_1-k_1}-1$. The variables x_k for $k = p_1+1,\ldots,p_1+p_2$ correspond to the stage-2 candidate factors, with $p_2 = 2^{(n_1+n_2)-(k_1+k_2)}-1$. And finally, the variables x_k for $k = p_1+p_2+1,\ldots,p_1+p_2+p_3$ correspond to the candidate factors for stage 3, with $p_3 = 2^{(n_1+n_2+n_3)-(k_1+k_2+k_3)}-1$.

Table 26: List of Constraints for 3-Staged FFSP Design

Constraint Type	Formula
(4.1.1)	$\sum_{k=1}^{p_1} x_k = n_1$
(4.1.2)	$\sum_{k=p_1+1}^{p_1+p_2} x_k = n_2$
(4.1.3)	$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} x_k = n_3$
(4.1.4)	$x_i + x_j - w_{ij} \le 1, 1 < i < j$
(4.1.5)	$x_i + x_j - 2w_{ij} \ge 0, 1 < i < j$
(4.1.6)	$\sum_{ij \in S^k} w_{ij} = y_k + z_k, \forall\, k$
(4.1.7)	$z_k \le M(1-y_k), \forall\, k$
(4.1.8)	$y_k + z_k \le M\tau_k, \forall\, k$
(4.1.9)	$(1-\tau_k)+x_k-\tau_k^{S1} \le 1, k=1,\ldots,p_1$
(4.1.10)	$(1-\tau_k)+x_k-2\tau_k^{S1} \ge 0, k=1,\ldots,p_1$
(4.1.11)	$\sum_{k=1}^{p_1} \tau_k^{S1} = ClearS1$
(4.1.12)	$(1-\tau_k)+x_k-\tau_k^{S2} \le 1, k=p_1+1,\ldots,p_1+p_2$
(4.1.13)	$(1-\tau_k)+x_k-2\tau_k^{S2} \ge 0, k=p_1+1,\ldots,p_1+p_2$

Constraint Type	Formula
(4.1.14)	$\sum_{k=p_1+1}^{p_1+p_2} \tau_k^{S2} = ClearS2$
(4.1.15)	$(1-\tau_k)+x_k-\tau_k^{S3} \leq 1, k = p_1+p_2+1,\ldots,p_1+p_2+p_3$
(4.1.16)	$(1-\tau_k)+x_k-2\tau_k^{S3} \geq 0, k = p_1+p_2+1,\ldots,p_1+p_2+p_3$
(4.1.17)	$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} \tau_k^{S3} = ClearS3$
(4.1.18)	$\sum_{ij\in S_{S1S1}^k} w_{ij} = y_k^{S1S1} + z_k^{S1S1}, k = 1,\ldots,p_1$
(4.1.19)	$z_k^{S1S1} \leq M(1-y_k^{S1S1}), k = 1,\ldots,p_1$
(4.1.20)	$y_k^{S1S1} + y_k + (1-x_k) - \tau_k^{S1S1} \leq 2, k = 1,\ldots,p_1$
(4.1.21)	$y_k^{S1S1} + y_k + (1-x_k) - 3\tau_k^{S1S1} \geq 0, k = 1,\ldots,p_1$
(4.1.22)	$\sum_{k=1}^{p_1} \tau_k^{S1S1} = S1S1\_2FIs$
(4.1.23)	$\sum_{ij\in S_{S1S2}^k} w_{ij} = y_k^{S1S2} + z_k^{S1S2}, k = p_1+1,\ldots,p_1+p_2$
(4.1.24)	$z_k^{S1S2} \leq M(1-y_k^{S1S2}), k = p_1+1,\ldots,p_1+p_2$
(4.1.25)	$y_k^{S1S2} + y_k + (1-x_k) - \tau_k^{S1S2} \leq 2, k = p_1+1,\ldots,p_1+p_2$
(4.1.26)	$y_k^{S1S2} + y_k + (1-x_k) - 3\tau_k^{S1S2} \geq 0, k = p_1+1,\ldots,p_1+p_2$
(4.1.27)	$\sum_{k=p_1+1}^{p_1+p_2} \tau_k^{S1S2} = S1S2\_2FIs$
(4.1.28)	$\sum_{ij\in S_{S1S3}^k} w_{ij} = y_k^{S1S3} + z_k^{S1S3}, k = p_1+p_2+1,\ldots,p_1+p_2+p_3$
(4.1.29)	$z_k^{S1S3} \leq M(1-y_k^{S1S3}), k = p_1+p_2+1,\ldots,p_1+p_2+p_3$
(4.1.30)	$y_k^{S1S3} + y_k + (1-x_k) - \tau_k^{S1S3} \leq 2, k = p_1+p_2+1,\ldots,p_1+p_2+p_3$
(4.1.31)	$y_k^{S1S3} + y_k + (1-x_k) - 3\tau_k^{S1S3} \geq 0, k = p_1+p_2+1,\ldots,p_1+p_2+p_3$
(4.1.32)	$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} \tau_k^{S1S3} = S1S3\_2FIs$
(4.1.33)	$\sum_{ij\in S_{S2S2}^k} w_{ij} = y_k^{S2S2} + z_k^{S2S2}, k = 1,\ldots,p_1+p_2$
(4.1.34)	$z_k^{S2S2} \leq M(1-y_k^{S2S2}), k = 1,\ldots,p_1+p_2$
(4.1.35)	$y_k^{S2S2} + y_k + (1-x_k) - \tau_k^{S2S2} \leq 2, k = 1,\ldots,p_1+p_2$
(4.1.36)	$y_k^{S2S2} + y_k + (1-x_k) - 3\tau_k^{S2S2} \geq 0, k = 1,\ldots,p_1+p_2$

Constraint Type	Formula
(4.1.37)	$\sum_{k=1}^{p_1+p_2} \tau_k^{S2S2} = S2S2\_2FIs$
(4.1.38)	$\sum_{ij \in S_{S2S3}^k} w_{ij} = y_k^{S2S3} + z_k^{S2S3}, k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3$
(4.1.39)	$z_k^{S2S3} \le M\left(1 - y_k^{S2S3}\right), k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3$
(4.1.40)	$y_k^{S2S3} + y_k + (1 - x_k) - \tau_k^{S2S3} \le 2, k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3$
(4.1.41)	$y_k^{S2S3} + y_k + (1 - x_k) - 3\tau_k^{S2S3} \ge 0, k = p_1 + p_2 + 1, \ldots, p_1 + p_2 + p_3$
(4.1.42)	$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} \tau_k^{S2S3} = S2S3\_2FIs$
(4.1.43)	$\sum_{ij \in S_{S3S3}^k} w_{ij} = y_k^{S3S3} + z_k^{S3S3}, \forall\, k$
(4.1.44)	$z_k^{S3S3} \le M\left(1 - y_k^{S3S3}\right), \forall\, k$
(4.1.45)	$y_k^{S3S3} + y_k + (1 - x_k) - \tau_k^{S3S3} \le 2, \forall\, k$
(4.1.46)	$y_k^{S3S3} + y_k + (1 - x_k) - 3\tau_k^{S3S3} \ge 0, \forall\, k$
(4.1.47)	$\sum_k \tau_k^{S3S3} = S3S3\_2FIs$
(4.1.48)	$x_{2^{m-1}} = 1, m = 1, \ldots, (n_1 + n_2 + n_3) - (k_1 + k_2 + k_3)$

Split-Split-Split-Split-Plot Example

Let's consider an experiment for a process involving 4 sequential stages (Ramirez & Tobias, 2007). The process has 1 factor at Stage 1, 4 factors at Stage 2, 2 factors at Stage 3, and 3 factors at Stage 4. A full factorial design would consist of $2^1 \times 2^4 \times 2^2 \times 2^3 = 1024$ runs. Clearly, this is an impractical number of experimental runs for most companies.

An initial attempt was performed to augment the design previously used for the 3-staged SPD to account for the additional stage. While the formulation of the new 4-staged IP model was feasible, computational time was impractical because we had constraints accounting for all types of 2FIs (S1S1, S1S2, S1S3, S1S4, S2S2, S2S3, S2S4,

S3S3, S3S4, S4S4). Therefore, we will revert back to a simpler IP model to compute this design. We will assume that for the purpose of this experiment, a Resolution IV design is desired, so that all main effects are clear and estimatable. The following set of constraints comprise the new simplified model for this 4-staged SPD:

$$\sum_{k=1}^{p_1} x_k = n_1 \tag{4.2.1}$$

$$\sum_{k=p_1+1}^{p_1+p_2} x_k = n_2 \tag{4.2.2}$$

$$\sum_{k=p_1+p_2+1}^{p_1+p_2+p_3} x_k = n_3 \tag{4.2.3}$$

$$\sum_{k=p_1+p_2+p_3+1}^{p_1+p_2+p_3+p_4} x_k = n_4 \tag{4.2.4}$$

$$x_i + x_j - w_{ij} \le 1, i < j \tag{4.2.5}$$

$$x_i + x_j - 2w_{ij} \ge 0, i < j \tag{4.2.6}$$

$$\sum_{i,j \in S_k} w_{ij} = y_k + z_k, \forall\, k \tag{4.2.7}$$

$$z_k \le M(1 - y_k), \forall\, k \tag{4.2.8}$$

$$y_k + z_k \le M(1 - x_k), \forall\, k \tag{4.2.9}$$

Constraints (4.2.1) – (4.2.4) set the number of factors in each stage. Constraints (4.2.5) – (4.2.8) are the same as type (2.2) and type (2.3) constraints from Chapter 2 for setting the variables representing 2FIs and counting the number of 2FIs aliased together for each letter-group. Finally, constraint (4.2.9) guarantees the design is Resolution IV or

greater. In order to find a design with the maximum number of clear 2FIs (y_k), we will minimize the number of aliased 2FIs, using the following objective function:

$$\text{Min } Z = \sum_k z_k \tag{4.2.10}$$

This objective function is equivalent to maximizing the sum of y_k, but uses less computational time in AMPL (approximately 20 seconds).

Applying this simplified model to the 4-staged SPD provides us with a Resolution IV design, where all the main effects are clear and there are 33 clear 2FIs. This confirms the design found by Ramirez and Tobias. By using an IP model, we could further add constraint to restrict the types of 2FIs that are clear. In Ramirez and Tobias, the design had three 2FIs between the stage 1 factor and 3 stage 2 factors confounded with other 2FIs. Let's say the experimenter was highly interested in the stage 1 factor and all of its 2FIs. We can attempt to guarantee these sought after effects are clear by adding the following customized constraints:

$$y_k + z_k + x_k \le M(1 - w_{1,s}) + 1,\ s = p_1 - 1, \ldots, p_1 + p_2 + p_3 + p_4 \tag{4.2.11}$$

$$y_k + z_k + x_k \le M(1 - w_{s,1}) + 1,\ s = p_1 - 1, \ldots, p_1 + p_2 + p_3 + p_4 \tag{4.2.12}$$

The resulting design still has 33 clear 2FIs, but the factor from stage 1 and all its 2FIs with other factors are now clear. This demonstrates the flexibility of generating custom split-plot designs using IP models.

CHAPTER 5: APPLYING A GENETIC ALGORTHIM TO A KRONECKER MATRIX TO GENERATE SPLIT-PLOT DESIGNS

This chapter focuses on some initial work our research committee performed involving the creation of FFSP designs using the Kronecker product operation and a genetic algorithm. While the direction of our research shifted to the integer programming method presented in earlier sections of this document, we feel it is beneficial to present our initial approach for possible future research.

Kronecker Product Operator

Murat Kulahci (2007) presents a flexible matrix representation for two-level fractional factorial designs to allow the user to block the experiment based on custom design criterion. He first points out the need for such a flexible matrix. Although there have been several authors to create tables for blocking fractional factorial designs up to 128 runs, these tables are generated using specific design criterion. However, with the increase in computing power, Dr. Kulahci proposes the future statistical software packages will have features allowing the experimenter to create custom design criterion.

To generate the flexible matrix for blocking, Kulahci uses the properties of the Kronecker product. The matrix representation of a 2^k factorial design can be generated using the Kronecker product. Dey and Mukerjee (1999) prove that the Kronecker product of two Hadamard matrices is also a Hadamard matrix.

The matrix representation of a two-level factorial design with k factors can be written as

$$2^k = \underbrace{2^1 \otimes 2^1 \otimes \cdots \otimes 2^1}_{k}$$

$$= 2^{j} \otimes 2^{k-j},\ 1 \le j \le k-1$$

where $\otimes$ is the Kronecker product operator and

$$2^{1} = \begin{bmatrix} 1 & -1 \\ 1 & 1 \end{bmatrix}$$

The 2^k design matrix is a 2^k by 2^k array where the columns correspond to the intercept, k main effects, and all interactions of the k effects. For $k=1$, the factor notation for this design matrix is

$$2^{1} = [I \quad A]$$

For $k=2$, an additional factor column is added to the matrix, along with the interactions with the columns already in the design matrix.

$$2^{2} = [I \ A \ B \ AB]$$

This process continues as k increases. For $k=4$,

$$2^{4} = [I \ A \ B \ AB \ C \ AC \ BC \ ABC \ D \ AD \ BD \ ABD \ CD \ ACD \ BCD \ ABCD]$$

Next, Kulahci shows how the columns of this design matrix can be partitioned into candidates for blocking factors and treatment variables in a blocked design. This idea is also applied to blocking fractional factorial designs. Kulahci identifies a few limitations as to which fractional designs can be used for blocking with various runs. Kulahci provides the initial stages for my proposal on designing split-plot designs using the matrix representation of the 2^k factorial design generated using the Kronecker product.

Chen and Cheng (2006) present a method of constructing two-level designs of Resolution IV. This method uses the Kronecker product to "double" a design. They prove that the double of a Resolution IV is also a Resolution IV design with twice as

many runs and factors. They also show that the projection of any subset of factors results in a design of Resolution IV or greater.

Genetic Algorithms

Choosing a set of columns from the Kronecker matrix can result in a complex global optimization problem. As the number of runs and factors increase, the numbers of possible solutions increase, and depending on the goal of the design, there can be very few designs that meet the optimization goal. An optimization heuristic algorithm is used to search through the solution space for the global optimum. One such heuristic is the genetic algorithm (GA). The pseudo-code algorithm for a GA is as follows:

1. Select initial population
2. Evaluate each individual of the population based on the scoring criteria
3. Repeat for each generation
    a. Select best-ranked individuals from population as parents
    b. Breed new generation through crossover and mutation
    c. Evaluate the new offspring
    d. Replace worst-ranked individuals of population with offspring
4. Until predetermined number of generations or termination goal is met

Creating the Split-Plot Candidate Set Using the Kronecker Product

A split-plot design with two subplots can be written in matrix notation as

$$\begin{bmatrix} w & -s \\ w & s \end{bmatrix}$$

Here, w represents a whole-plot factor and s represents the subplot factor. This is only one whole plot, and we must change the whole-plot factor and rerun the subplots. Using the Kronecker product operator, we get

$$\begin{array}{c} \begin{array}{cccc} I & A & B & AB \end{array} \\ \begin{bmatrix} 1 & -s & -w & s \\ 1 & s & -w & -s \\ 1 & -s & w & -s \\ 1 & s & w & s \end{bmatrix} \end{array}$$

Column 1 represents the intercept and column 3 represents the one whole-plot factor. Columns 2 and 4 are used for the subplot factor. We can also use columns 2 and 4 to represent 2 subplot factors.

We will designate a split-plot design as $2^i \otimes 2^j$, where 2^i is the number of whole-plots with each whole-plot having 2^j subplots. This forms a 2^{i+j} by 2^{i+j} Hadamard matrix, with the first column being a column of 1's representing the intercept. The possible columns for the whole-plot factors are

$$c*\left(\frac{2^i}{i}\right)+1\,, \qquad c=1,\ldots,2^{i-1}-1 \tag{5.1}$$

The remaining $\left(2^j-1\right)\times 2^i$ columns are possible choices for subplot factors.

Each time we apply the Kronecker product operator to the matrix, we are essentially adding a factor and all its interactions with the original design matrix. This is also referred to as Yates order. Let's go from 2^2 to 2^3. We add basic factor C and the interactions with C.

$$
\begin{array}{cccccccc}
I & A & B & AB & C & AC & BC & ABC \\
\end{array}
$$

$$
\begin{bmatrix}
1 & -1 & -1 & 1 & -1 & 1 & 1 & -1 \\
1 & 1 & -1 & -1 & -1 & -1 & 1 & 1 \\
1 & -1 & 1 & -1 & -1 & 1 & -1 & 1 \\
1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\
1 & -1 & -1 & 1 & 1 & -1 & -1 & 1 \\
1 & 1 & -1 & -1 & 1 & 1 & -1 & -1 \\
1 & -1 & 1 & -1 & 1 & -1 & 1 & -1 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 1
\end{bmatrix}
$$

To obtain four basic factors, we would add D and all of its interactions; thus, completing the full 2^4 factorization.

Let's look at the 16-run split-plot design. With only 2 whole-plots, there will be 8 subplot runs in each whole plot. Thus, we have a $2^1 \otimes 2^3$ Kronecker matrix partitioned as follows.

I	S	S	S	S	S	S	S	W	S	S	S	S	S	S	S
1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1
1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1
1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
1	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1
1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1
1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Equation (5.1) shows that column 9 is the only possible column for the WP factor. Note that the value for column 9 stays constant within the 8 runs of the first whole-plot and the 8 runs of the second whole-plot. If we want more than two whole plots, we can use the same matrix, but the partition will be different.

I	S	S	S	W	S	S	S	W	S	S	S	W	S	S	S
1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1
1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1
1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
1	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1
1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1
1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

This is the $2^2 \otimes 2^2$ design with four whole-plots. We can have up to 3 WP factors, but if there are 3, then it will be a Res III design. Most likely we will have only 2 WP factors. The possible choices for WP factor columns, using equation (5.1), are $\{5,9,13\}$. And using the basis factor approach, we know that the first two WP factor columns are columns 9 and 5. These represent the first two basic columns in the 2^4 full factorial design. Together with column 13, these are the only columns that have values that stay the same during each whole-plot. The remaining columns are choices for SP factors, sans the identity column. Plus, we know that the first two SP factor columns will be columns 3 and 2, representing the other two basic factor columns in the 2^4 full factorial design.

Finally, let's consider the $2^3 \otimes 2^1$ partition with 8 whole-plots, each having 2 subplot runs.

I	S	W	S	W	S	W	S	W	S	W	S	W	S	W	S
1	-1	-1	1	-1	1	1	-1	-1	1	1	-1	1	-1	-1	1
1	1	-1	-1	-1	-1	1	1	-1	-1	1	1	1	1	-1	-1
1	-1	1	-1	-1	1	-1	1	-1	1	-1	1	1	-1	1	-1
1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1	1	1	1	1
1	-1	-1	1	1	-1	-1	1	-1	1	1	-1	-1	1	1	-1
1	1	-1	-1	1	1	-1	-1	-1	-1	1	1	-1	-1	1	1
1	-1	1	-1	1	-1	1	-1	-1	1	-1	1	-1	1	-1	1
1	1	1	1	1	1	1	1	-1	-1	-1	-1	-1	-1	-1	-1
1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1	1	-1
1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1	1	1
1	-1	1	-1	-1	1	-1	1	1	-1	1	-1	-1	1	-1	1
1	1	1	1	-1	-1	-1	-1	1	1	1	1	-1	-1	-1	-1
1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1
1	1	-1	-1	1	1	-1	-1	1	1	-1	-1	1	1	-1	-1
1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1	1	-1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

The column choices for WP factors are $\{3,5,7,9,11,13,15\}$. The first three WP factors will be the basis columns 9, 5 and 3. There can be up to 7 WP factors, forming a saturated WP design and poor alias structure. The remaining 8 columns can be designated as SP factors.

Another method for generating a Resolution IV split-plot design is to first use the doubling and projection method of Chen and Cheng (2006) to produce the flexible matrix of candidate columns. Let K be an initial matrix consisting of two factors and four runs.

$$K = \begin{bmatrix} 1 & -1 \\ 1 & 1 \\ -1 & -1 \\ -1 & 1 \end{bmatrix}$$

Note K is a Resolution IV design. By applying the doubling method four times, we arrive at a matrix with 64 rows and 32 columns. We now have a 64-run design with 32 factors of Resolution IV. Rearrangement of the run order does not affect the resolution of the design; therefore, we can rearrange the rows to match the blocked form of a split-plot design. The new matrix represents half of the columns from the SPD using the Kronecker method described above. Using the $2^4 \otimes 2^2$ notation to represent a SPD with 16 whole-plots, each with 4 subplot runs, our doubling matrix now provides us with 8

columns to choose the 7 WP factors, with the remaining 24 columns for the 8 SP factors. The remainder of this section focuses on the Kronecker method previously described for generating the matrix of candidate columns, and will not look at the alternative approach of doubling the design matrix.

Using Genetic Algorithm to Search for an Optimal Design

Regardless of the method to produce the candidate matrix, we are still left with two collections of candidate columns to select the remaining whole-plot and subplot factor columns. We apply a genetic algorithm search heuristic to search through the design space for the optimal solution. We first generate an initial population by randomly selecting the whole-plot and subplot factor columns from the candidate list of the Kronecker matrix. The initial population is set at 1000. We evaluate the initial population using the predetermined method of design evaluation criterion. For example, we may be simply counting the number of clear two-factor interactions. Once all members of the population have been scored based on this criterion, we select the top 10% of the designs to form the parents. We then generate the next generation of offspring using the parents. Two parents are selected at random. These two parents will generate two offspring. The set of whole-plot factors from one parent is combined with the set of subplot factors from the other parent. This generates one offspring. Combining the remaining factors from these two parents generate the other offspring. This procedure is repeated, randomly selecting two parents and generating two offspring, until the second generation population is created. The offspring comprising the second generation is scored based on the criterion and ranked along with the initial set of parents. The top

10% from this second generation will now form the new set of parents. This procedure is repeated over 25 generations. Thus, we have evaluated a total of 1000 x 25 = 25,000 design matrices. This is a fraction of the total solution space.

Designs With More Than 2 Steps

The Kronecker matrix can be used to generate the design for a split-plot design with *n* number of steps. There exists one whole plot design, and *n*-1 subplot designs, with each whole plot design nested within the others. Let's look at a basic example as to how this approach can be implemented. Consider the 64-run Kronecker matrix. Let there be a three-stage process with 2 factors in the first stage, 3 factors in the second stage, and 4 factors in the last stage. We can look at this as a $2^2 \otimes 2^2 \otimes 2^2$ design. This notation represents a whole-plot design consisting of 4 whole plots. Within each whole-plot, there exists a sub split-plot design with the other two stages. The 16 runs within each whole-plot can be viewed as a $2^2 \otimes 2^2$ design. Thus there are 4 plots for the second stage, with each plot consisting of four runs.

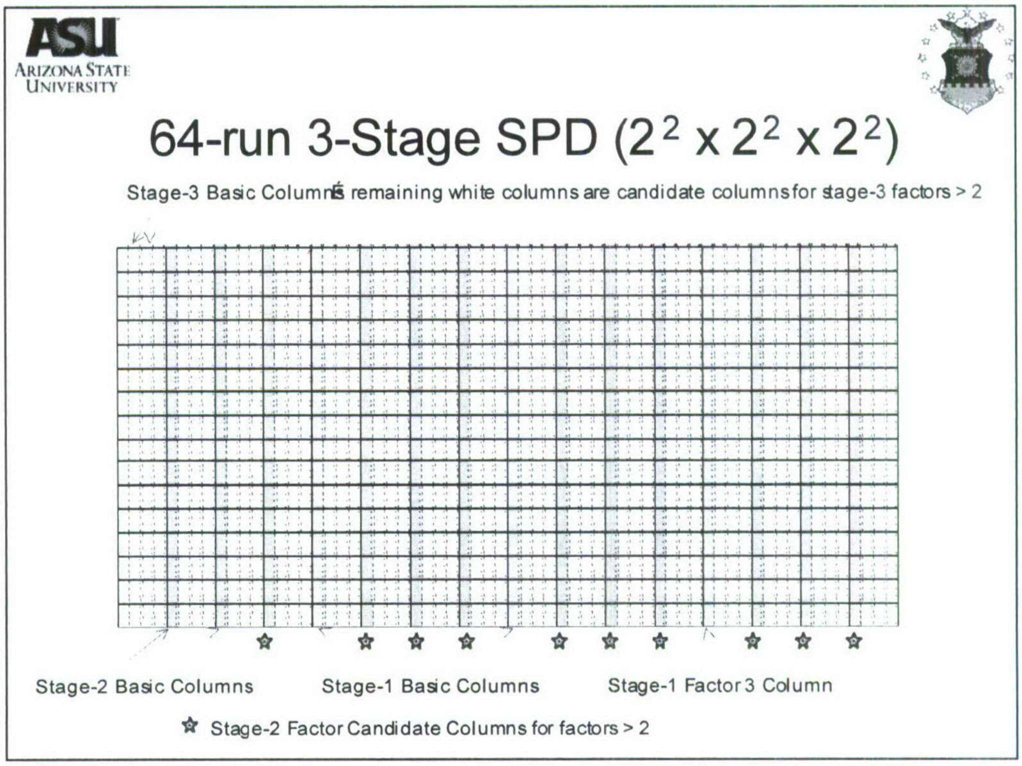


Figure 1: Matrix partitioning scheme for a 64-run 3-staged SPD

Expanding from the proposed approach for a two-stage split-plot design, we can see that the first stage has 3 possible candidate columns (*e, f, ef*) , the second stage has 12 candidate columns (*c, d, cd, and interaction columns with stage-1 candidate columns*), while the final stage of the design has the remaining 48 columns to select for factors (*a, b, ab, and interactions with columns from stage-1 and stage-2 columns*) . Again, using the same principle we applied to the two-staged split-plot design, we can already identify 2 basic factor columns for each stage due to the design we are considering. Thus, there remains one factor to designate for stage 2 and two factors to designate in stage 3. Determining the selection for these remaining factors is once again a choice based on the desired characteristics of the final design. Exploration of this design example and a

generalization of the n-staged split-plot design using the Kronecker matrix could be explored in future research.

Example 1: $2^4 \otimes 2^2$ 7/8 SPD

Let's look at a two-level design with 64 runs with 7 whole-plot factors and 8 subplot factors. using 16 whole plots, each with 4 subplot runs ($2^4 \otimes 2^2$). This is the same example design provided by Kulahci et al. (2006). The flexible matrix provided by the Kronecker product leads to 15 candidate columns for the 7 whole-plot factors and 48 candidate columns for the 8 subplot factors, resulting in over 2.4 Trillion possible balanced split-plot designs. An exhaustive search of all designs for any custom criterion is not feasible unless some preprocessing occurs.

The first phase of preprocessing requires a little bit of explanation. Let's step back and look at a 2^4 balanced two-level factorial design. In order to have 16 distinct runs, we must have the four basic factor columns $[A \;\; B \;\; C \;\; D]$. Let's say we select four arbitrary independent columns, $[A \;\; AB \;\; C \;\; D]$. Rearranging the order of the runs will still produce the four basic factor columns. For a fractional factorial design, 2^{m-n}, where $m-n=k$, we know there must be k basic factor columns. Thus, the $2^{7-3}=2^4$ fractional factorial design has $[A \;\; B \;\; C \;\; D]$ as the basic factor columns and three more chosen from the interactions of these basic columns. Creating a split-plot design using the flexible matrix described above, $2^i \otimes 2^j$, we see that there are a total of $i+j$ basic factor columns, i of which belong to whole-plot factors and the other j basic factor columns will be used as subplot factors.

Going back to our example with $2^4 \otimes 2^2$, we know that the first 4 whole-plot factors will be the four basic factor columns $[C \ D \ E \ F]$ using the Yates order notation described in a previous section. The other two basic factor columns $[A \ B]$ are the first two subplot factors. After assigning these six basic factor columns, there are 3 remaining whole-plot factors to choose from the remaining 11 candidate columns. Similarly, there are 6 remaining subplot factors to choose from the remaining 46 candidate columns. This reduces the number of possible designs to 1.5 Billion.

Now assume we want to have clear main effects in addition to maximizing clear two-factor interactions as our custom criterion, i.e Resolution IV design. We can perform a second preprocessing step to further reduce the solution space. We have already shown that the six basic factor columns are in the solution. We can therefore eliminate the columns representing the two-factor interactions between these basic factors from the candidate lists.

$$2^4 \otimes 2^2 = [I \quad A \quad B \quad \cancel{AB} \quad C \quad \cancel{AC} \quad \cancel{BC} \quad ABC \quad D \quad \cancel{AD} \quad \cancel{BD} \quad ABD \quad \cancel{CD} \quad ACD \quad BCD \quad \ldots]$$

This eliminates 15 more candidate columns: 6 from the whole-plots and 9 from the subplots. Now there are only 23 Million possible designs. Using an exhaustive search, the Matlab code requires just over 70 minutes finding the maximum number of clear two-factor interactions for Resolution IV designs. There are 27 clear two-factor interactions, and this solution is the same as that used in Kulahci et al. (2006). Using the simple GA algorithm, the same solution was found in less than a minute. Next, we consider Resolution IV designs and count the number of clear *cross* two-factor interactions. That is, the two-factor interactions (2FI) with one whole-plot factor and one subplot factor.

The exhaustive search took over two-hours and resulted in 14 clear cross 2FI. These 14 clear interactions are all cross 2FIs consisting of two distinct subplot factors. The GA algorithm took less than 45 seconds and found a different set of columns from the Kronecker matrix; however, the result still had all cross 2FI dealing with two subplot factors clear. Since the run order for the subplot runs with each whole-plot can be rearranged without changing the alias structure, this solution can be considered the same as that found using the exhaustive search.

For a Resolution III design, the exhaustive search has not been completed at this time due to the time needed to compute all 1.5 Billion designs. The GA method took less than a minute and resulted in 32 clear 2FIs. For the maximum number of clear cross 2FIs, the GA algorithm produced a design with 26 in less than a minute.

Example 2: $2^4 \otimes 2^2$ 5/4 SPD with custom criterion

The maximum number of clear 2FIs or clear cross 2FIs are traditional design criterion. We want to allow the user to select specific main effects and/or two-factor interactions to be clear of other effects. This initial study allows the user to specify the number of whole-plot and subplot factors that must have all of their two-factor interactions clear. For example, using the $2^4 \otimes 2^2$ matrix, let's have 5 WP and 4 SP factors. Due to the system under investigation, there may be 3 important WP factors and 2 important SP factors that the experimenter wishes to remain clear, even their two-factor interactions. Our custom GA searches for designs with the most clear 2FIs involving the first three WP factors and first two SP factors. We can look at the "first" factors because the experiment can still assign the important variables to these factors before running the

experiment. The GA code took less than a second to solve several versions of this example (various number of important WP and SP factors), both for Resolution III and IV designs.

It must also be noted that when running this code looking for a Resolution III design, the initial population did not contain the optimal answer. Thus, the GA approach is improving the solution over the generations.

This initial study only looked at maximizing the clear 2FIs with these important main effects. Future research could allow the user to further customize the criterion, including specific main effects and specific 2FIs.

Other Considerations

The majority of these GA runs found the optimal solution generating the initial population of 1000 random designs. So for the $2^4 \otimes 2^2$ 7/8 Res IV SPD, with over 23 Million design options, there must be a considerable number of designs that have 27 clear 2FIs. These designs could actually be considered equivalent designs once the whole-plots or runs within each whole-plot are rearranged. Further preprocessing could be performed to remove the redundant design in the solution space.

CHAPTER 6: CONCLUSIONS, SUMMARY OF ORIGINAL CONSTRIBUTIONS, AND FUTURE RESEARCH

Conclusions

In Chapter 2, fractional factorial split-plot designs were represented using Yates order numbering scheme. Using this numbering scheme, a integer programming model was developed to create optimal FFSP designs according various performance criteria, including maximum number of clear 2FIs. Using these IP models, both Resolution III and Resolution IV FFSP designs were generated for 8-, 16-, and 32-runs with various numbers of whole-plot and subplot factors. By incorporated the numbering scheme for the main effects and design generators, additional IP constraints were derived to customize the design further. Using these constraints, an experimenter can search for designs that isolate the effects of individual main effects and their corresponding 2FIs. This customized approach was also demonstrated on the example designs provided. It was shown how the FFSP designs generated, which are *D*-optimal, compare to the *D*-optimal split-plot designs generated by using common statistical software. The IP model allows the experimenter to create customized designs with main effects and 2FIs clear of other main effects and 2FIs, while the statistical software only allows the experimenter to require these effects to be estimatable.

In Chapter 3, blocking FFSP designs was introduced. In order to incorporate the same numbering scheme as in Chapter 2, a new approach for represented blocked designs was presented. Blocking effects are now considered main effects, or *basic factors*. Design generators for the whole-plot design and subplot designs are generated using a combination of *basic factor* letters from the blocking factors and whole-plot/subplot

basic factors. Once incorporated into the numbering scheme, the IP model from Chapter 3 was modified to generate blocked FFSP designs. The chrome-plating case study provided an opportunity to compare our IP generated design with a minimum aberration design and a robust parameter design. Other examples provided evidence that our IP model can generate blocked FFSP designs that meet or exceed the performance characteristics of those presented in current literature.

Chapter 4 expanded the IP model to generate FFSP designs for sequential processes with more than 2 stages. The model formulation was provided for a three-staged process, so that all types of 2FIs were accounted for and could be incorporated into the objective function accorded to the experimenter's needs. A split-split-split-split plot example (4-stages) was provided to show how the IP model could still be augmented with customized constraints to isolate individual main effects and their interactions.

Although our initial research shifted to using IP models to create FFSP designs, some initial study was conducted on applying modern heuristic techniques to a candidate matrix of design runs to create the split-plot designs. Chapter 5 provides a detailed overview of this research proposal.

Original Contributions

A new IP model was developed to generate FFSP designs. The model formulation allows the experimenter to customize the alias structure.

A numbering scheme based on the letter-group notation for the design generators was developed in order to linearize the constraints on the model. Representing the letter-groups as binary variables provided the means to linearize the constraints on the model to meet the IP requirements. Constraints were created to dictate the aliasing structure of the

first order effects and their interactions. By customizing these constraints, FFSP designs were generated with specific main effects and 2FIs clear of any confounding effects.

A model was also developed to create blocked split-plot designs. By considering the blocked effects as *basic factors*, we are able to adjust the IP model constraints to ignore Blocking x Blocking effects and Blocking x WP/SP factor effects when calculating the number of clear 2FIs. It was shown how this model, along with flexibility in the constraints and objective function, can generate split-plot designs that exceed the performance of competing literature examples.

By expanding this IP model, customized split-plot designs were created for manufacturing processes with more than two steps. The customizable objective function allows the experimenter to optimize specific types of clear effects and 2FIs. Customizable constraints can also be added to isolate individual main effects and their interactions. While there exists some software available today to create FFSP designs for multiple stages (SAS), the user does not yet have the capability to customize the design to the extent that we have shown.

Genetic Algorithms were extended for use in the creation of split-plot designs from a design matrix representing a candidate set of factor columns. The Kronecker product operator was used to create the matrix, and a partitioning scheme was developed to separate the whole-plot factor candidates from the subplot factor candidates.

Future Research

In this work, it was assumed that interactions of order greater than 2 were negligible. The model representation introduced could be extended for the situation where three-factor interactions (3FIs) are also considered; however, advances in computer speeds

would be necessary do to the added complexity by having 3FIs in the model. Chapter 3 looked as modifying the IP model to generate blocked FFSP designs, and several examples were explored. The models considered in this chapter were not customized using the customized constraints as shown in Chapter 2. Further consideration could explore blocked designs with customized constraints to target individual effects and interactions.

Chapter 5 provided a general approach for using genetic algorithms to search for FFSP designs among the columns of the Kronecker matrix. While initial results were promising in finding solutions, it was unclear as to how much the heuristic played a role in the final solution. It would be useful to explore various strategies for generating the initial GA population and the mutation scheme to see if further improvements could be made on the designs.

REFERENCES

Almimi, A. A., Kulahci, M., & Montgomery, D. C. (2008). Follow-up Designs to Resolve Confounding in Split-Plot Experiments. *Journal of Quality Technology* , to appear.

Bingham, D. R., & Sitter, R. R. (2001). Design Issues in Fractional Factorial Split-Plot Experiments. *Journal of Quality Technology , 33*, 2-15.

Bingham, D. R., & Sitter, R. R. (1999). Minimum-Aberration Two-Level Fractional Factorial Split-Plot Designs. *Technometrics , 41*, 62-70.

Bingham, D. R., & Sitter, R. R. (1999). Some Theoretical Results for Fractional Factorial Split-Plot Designs. *The Annals of Statistics , 27*, 1240-1255.

Bisgaard, S. (2000). The Design and Analysis of 2^(k-p) x 2^(q-r) Split-Plot Experiments. *Journal of Quality Technology , 32*, 39-56.

Box, G. P., & Hunter, J. S. (1961). The 2^(n-p) Fractional Factorial Designs Part I. *Technometrics , 3*, 311-352.

Box, G. P., & Hunter, J. S. (1961). The 2^(n-p) Fractional Factorial Designs Part II. *Technometrics , 3*, 449-458.

Box, G. P., & Jones, S. (1992). Split-Plot Designs for Robust Product Experimentation. *Journal of Applied Statistics , 19*, 3-26.

Box, G. P., Hunter, W. G., & Hunter, J. S. (2001). *Statistics for Experimenters.* New York: Wiley.

Capehart, S., Keha, A., Kulahci, M., & Montgomery, D. (2008). Designing Fractional Factorial Split-Plot Experiments Using Integer Programming. *Technometrics* , submitted.

Chen, B., Li, P., Liu, M., & Zhang, R. (2006). Some Results on Blocked Regular 2-Level Fractional Factorial Designs with Clear Effects. *Journal of Statistical Planning and Inferece , 136*, 4436-4449.

Chen, H., & Cheng, C. (2006). Doubling and Projection: A Method of Constructing Two-Level Designs of Resolution IV. *The Annals of Statistics , 34*, 546-558.

Chen, J. (1992). Some Results on 2^(n-k) Fractional Factorial Designs and Search for Minimum Aberration Designs. *The Annals of Statistics , 20*, 2124-2141.

Chen, J., Sun, D. X., & Wu, C. F. (1993). A Catalogue of Two-Level and Three-Level Fractional Factorial Designs with Small Runs. *International Statistical Review , 61*, 131-145.

Chen, J., Sun, D., & Wu, C. (1993). A Catalogue of Two-Level and Three-Level Fractional Factorial Designs with Small Runs. *International Statistical Review , 61*, 131-145.

Cheng, C.-S., Steinberg, D. M., & Sun, D. X. (1999). Minimum Aberration and Model Robustness for Two-Level Fractional Factorial Designs. *Journal of the Royal Statistical Society , 61*, 85-93.

Dey, A., & Mukerjee, R. (1999). *Fractional Factorial Plans.* New York: Wiley.

Franklin, M. F., & Bailey, R. A. (1977). Selection of Defining Contrasts and Confounding Effects in Two-Level Experiments. *Applied Statistics , 26*, 321-326.

Fries, A., & Hunter, W. G. (1980). Minimum Aberration 2^(k-p) Designs. *Technometrics , 22*, 601-608.

Goos, P. (2006). Optimal Versus Orthogonal and Equivalent-Estimation Design of Blocked and Split-Plot Experiments. *Statistica Neerlandica , 60*, 361-378.

Goos, P. (2002). *The Optimal Design of Blocked and Split-Plot Experiments.* New York: Springer.

Goos, P., & Vandebroek, M. (2001). D-Optimal Response Surface Designs in the Presence of Random Block Effects. *Computational Statistics and Data Analysis , 37*, 433-453.

Goos, P., & Vandebroek, M. (2001). Optimal Split-Plot Designs. *Journal of Quality Technology , 33*, 436-450.

Goos, P., & Vandebroek, M. (2004). Outperforming Completely Randomized Designs. *Journal of Quality Technology , 36*, 12-37.

Hadamard Matrix. (2005, November 15). Retrieved from Wolfram MathWorld: http://mathworld.wolfram.com/hadamardmatrix.html

Hamada, M., & Wu, C. J. (1992). Analysis of Designed Experiments with Complex Aliasing. *Journal of Quality Technology , 24*, 130-137.

Hawkins, D. M. (1993). Regression Adjustment for Variables in Multivariate Quality Control. *Journal of Quality Technology , 25*, 170-182.

Huang, P., Chen, D., & Voelkel, J. O. (1998). Minimum-Aberration Two-Level Split-Plot Designs. *Technometrics , 40*, 314-326.

Jacroux, M. (2006). Optimal Sequencing of Blocked 2^(m-k) Fractional Factorial Designs. *Journal of Statistical Planning and Inference , 136*, 1473-1492.

Jones, B., & Goos, P. (2007). A Candidate-Set-Free Algorith for Generating D-Optimal Split-Plot Designs. *Journal of the Royal Statistical Society , 56*, 347-364.

Kempthorne, O. (1998). *The Design and Analysis of Experiments.* New York: John Wiley & Sons.

Kulahci, M. (2007). Blocking Two-Level Factorial Experiments. *Quality and Reliability Engineering International , 23*, 283-289.

Kulahci, M., Ramirez, J., & Tobias, R. (2006). Split-Plot Fractional Designs: Is Minimum Aberration Enough. *Journal of Quality Technology , 38*, 56-64.

Liang, L., Anderson-Cook, C. M., & Robinson, T. J. (2006). Fraction of Design Space Plots for Split-Plot Designs. *Quality and Reliability Engineering International , 22*, 275-289.

Liao, C., & Iyer, H. (1999). A Stochastic Algorithm for Selecting of Defining Contrasts in Two-Level Experiments. *Biometrical Journal , 41*, 671-678.

Lin, D. K. (1993). Another Look at First-Order Saturated Designs: The P-Efficient Designs. *Technometrics , 35*, 282-292.

McLeod, R. G. (2007). Optimal Block Sequences for Blocked Fractional Factorial Split-Plot Designs. *Journal of Statistical Planning and Inference* , doi:10.1016/j.jspi.2007.10.012.

McLeod, R. G., & Brewster, J. F. (2006). Blocked Fractional Factorial Split-Plot Experiments for Robust Paramter Design. *Journal of Quality Technology , 38*, 267-280.

McLeod, R. G., & Brewster, J. F. (2004). The Design of Blocked Fractional Factorial Split-Plot Experiments. *Technometrics , 46*, 135-146.

Mee, R., & Bates, R. (1998). Split-Plot Designs: Experiments for Multistage Batch Processes. *Technometrics , 40*, 127-140.

Milliken, G. A., Shi, X., Mendicino, M., & Vasudev, P. K. (1998). Strip-Plot Design for Two-Step Process. *Quality and Reliability Engineering International , 14*, 197-210.

Montgomery, D. C. (2001). *Design and Analysis of Experiments* (6th ed.). New York: Wiley.

Montgomery, D. C., Borror, C. M., & Stanley, J. D. (1997). Some Cautions in the Use of Plackett-Burman Designs. *Quality Engineering , 10*, 371-381.

Myers, R. H., & Montgomery, D. C. (2002). *Response Surface Methodology.* New York: John Wiley & Sons.

Parker, P. A., Kowalski, S. M., & Vining, G. G. (2006). Classes of Split-Plot Surface Designs for Equivalent Estimation. *Quality and Reliability Engineering International , 22*, 291-305.

Perry, L. A., Montgomery, D. C., & Fowler, J. W. (2001). Partition Experimental Designs for Sequential Processes: Part I-First Order Models. *Quality and Reliablity Engineering International , 18*.

Perry, L. A., Montgomery, D. C., & Fowler, J. W. (2002). Partition Experimental Designs for Sequential Processes: Part II-Second Order Models. *Quality and Reliability Engineering International , 18*, 373-382.

Plackett, R. L., & Burman, J. P. (1946). The Design of Optimum Multifactorial Experiments. *Biometrika , 33*, 328-332.

Raktoe, B. (1976). On Alias Matrices and Generalized Defining Relationships of Equi-information Factorial Arrangements. *Journal of the Royal Statistical Society , 38*, 279-283.

Ramirez, J. G., & Tobias, R. (2007). Split and Conquer! Using SAS/QC to Design Quality into Complex Manufacturing Processes. *SAS Global Forum.*

Vining, G., Kowalski, S., & Montgomery, D. (2005). Response Surface Designs Within a Split-Plot Structure. *Journal of Quality Technology , 37*, 115-129.

Wu, C. J., & Hamada, M. (2000). *Experiments: Planning, Analysis, and Parameter Design Optimization.* New York: John Wiley & Sons.

Wu, H., & Wu, C. J. (2002). Clear Two-Factor Interactions and Minimuum Aberration. *Annals of Statistics , 30* (5).

Yang, J., Li, P., Liu, M., & Zhang, R. (2006). 2^(n1+n2)-(k1+k2) Fractional Factorial Split-Plot Designs Containing Clear Effects. *Journal of Statistical Planning and Inference , 136*, 4450-4458.

Yates, F. (1935). Complex Experiments. *Supplement to the Journal fo the Royal Statistical Society , 2*, 181-247.

Zhang, R., & Park, D. (2000). Optimal Blocking of Two-Level Fractional Factorial Designs. *Journal of Statistical Planning and Inference , 91*, 107-121.